Chaos in the Laboratory

and 13 Other Science Projects Using the Apple® II

Edited by David L. Vernier

Contributing Authors:
J. Roger Bacon
Peter Bruecken
Mark C. Helmlinger
Satoshi Kino
David L. Vernier

Vernier
SOFTWARE

Chaos in the Laboratory

and 13 Other Science Projects Using the Apple® II

Edited by David L. Vernier

Contributing Authors:

J. Roger Bacon
Peter Bruecken
Mark C. Helmlinger
Satoshi Kino
David L. Vernier

Fourteen educational projects for making measurements and controlling electrical devices through the Apple II+/IIe/IIGS game port. Includes two disks of sample programs.

Chaos in the Laboratory

and 13 Other Science Projects Using the Apple® II

Edited by David L. Vernier

Contributing Authors:

J. Roger Bacon
Peter Bruecken
Mark C. Helmlinger
Satoshi Kino
David L. Vernier

Vernier Software
2920 S.W. 89th Avenue
Portland, Oregon 97225
(503) 297-5317
(503) 297-1760

Chaos in the Laboratory and 13 Other Science Projects Using the Apple® II. Copyright © 1991 by Vernier Software. All rights reserved. No part of this book may be used or reproduced in any manner whatsoever without written permission of the publisher except in the case of brief quotations embodied in critical articles or reviews.

Every effort has been made to supply complete and accurate information. However, Vernier Software assumes no responsibility for its use, nor for any infringements of patents or other rights of third parties that would result. Neither is any liability assumed for damages resulting from the use of the information contained herein.

Apple, Apple II, Applesoft, Apple IIGS, Apple II+, Apple IIe, Apple IIc, Apple IIc+, ImageWriter, LaserWriter, DOS 3.3, ProDOS, AppleWorks, FILER and CONVERT are either registered trademarks, trademarks, or copyrighted by Apple Computer, Inc.

Published by
Vernier Software
2920 S.W. 89th Avenue
Portland, OR 97225
(503) 297-5317

Photographs by Photo Tek, Hillsboro, Oregon

Library of Congress Catalog Card Number: 91-090900
ISBN 0-918731-46-1
First Edition 1991
First Printing
Printed in the United States of America

Dedicated to Vernier Software customers, who for ten years have provided us with ideas, suggestions, constructive criticism, pats on the back, and an income.

About the Authors

David L. Vernier is a former high school science teacher with a life-long interest in science projects. He is the author of the popular book, *How to Build A Better Mousetrap and Thirteen Other Projects Using the Apple® II*. He also develops science software and laboratory interfacing hardware for the Apple II, IBM, and Macintosh computers.

J. Roger Bacon is a professor of chemistry at Western Carolina University, where he teaches analytical and environmental chemistry. Roger has had an interest in electronics and instrumentation since graduate school. In addition to his college teaching, Roger has conducted several workshops for high school teachers and students involving computer interfacing and environmental chemistry.

Peter Bruecken is a high school physics teacher in Bettendorf, Iowa. He has been a leader of numerous summer teacher training programs and is a 1986 Physics Teacher Resource Agent (PTRA).

Mark Helmlinger is a student at Cal Poly San Luis Obispo and a laboratory technician at Mt. San Antonio Community College. He is a physics major with special interest in physics demonstrations and chaos.

Satoshi Kino is an engineering student at University of Washington in Seattle, Washington. His project, Computer Vision, began as a high school science project, which eventually made it to the national level in science competitions by his senior year.

Acknowledgments

Many people helped in making this book a reality. Special thanks to all of the following:

Our many editors helped us immensely. Christine Vernier read every chapter over and over again and was the final builder of all the projects. She has endured long hours in front of the Macintosh and with her soldering iron. Other invaluable editors included: Richard Warner, Janet L. White, Lowell Herr, Walter Hellman, Chris Laufer, Dewey Beadle, and Frank Granshaw.

Almost every Vernier Software employee and anyone else who walked in the door over the last three years have been drafted into helping us test and edit these projects. Tom Wesselman wrote much of the Magnetic Field Sensor and D-to-A / Digitized Sound Project. Glen Gaidos helped to developed the Light Meter, Barometer, and Speed of Sound projects. Michael Mendenhall worked many hours on the Quiz Show project and the D-to-A Converter. Thanks also to Charles Nguyen, Kathy Pikkarainen, Patricia Ekman, Todd Bates, Mark Topinka, Todd Bates, Johan Hammerstrom, and Dan Christensen.

We feel it is essential that each of these projects be built by several test assemblers ranging from physics and electronics experts to complete novices. Thanks to all of them.

Many of the programs on the *Chaos in the Laboratory Programs* disks were written by Larry Arbuthnot and Tom Wesselman.

Matt Zimmerman of ZYX Measurement provided invaluable advice on electronic circuits.

Contents

Introduction xi

The *Chaos in the Laboratory Programs* Disks xiv
Electronic Components xvi
Sources of Electronic Components xxi
How to Solder xxiii
Other Project Assembly Tips xxvi
Troubleshooting xxvii
The Apple II Game Port xxviii
A Simple Game Port Interfacing Example xxxiii

Project 1 Thermistor Temperature Probe 1-1
Project 2 Quiz Show 2-1
Project 3 Computer Vision 3-1
Project 4 Colorimeter 4-1
Project 5 A-to-D Converter 5-1
Project 6 Light Meter 6-1
Project 7 Barometer/Pressure Sensor 7-1
Project 8 Magnetic Field Sensor 8-1
Project 9 Smart Battery Charger 9-1
Project 10 D-to-A/Digitized Sound 10-1
Project 11 Stepper Motors 11-1
Project 12 Speed of Sound 12-1
Project 13 Ultrasonic Range Finder 13-1
Project 14 Chaos in the Laboratory 14-1

Appendixes

A. Cables to the Apple Game Ports A-1
B. Resistor Color Codes B-1
C. Bibliography C-1
D. Software on the *Chaos in the Laboratory Programs* Disks D-1
E. Working with the Assembly-Language Programs E-1
F. Teacher's Guide F-1
G. Other Vernier Software Products G-1

Parts Kit Order Form

Feedback Form

Introduction

In 1986 we published a book of science projects for use with the Apple II computer. That book, *How to Build A Better Mousetrap and Thirteen Other Science Projects Using the Apple® II,* has turned out to be popular with both students and teachers. It has proven to be a good way to learn about electronics, computers, and science. It has been used in college and high school classes and in several teacher workshops. Most importantly, it has been the start of a number of great science projects.

This is a book of fourteen different projects with a similar intent. All of the projects involve *laboratory interfacing*; that is, using the computer as a laboratory instrument. They all involve construction of a circuit that connects to the Apple II game port and allows the computer to interact with the outside world. The projects are designed to be done by students (or teachers) with minimal experience in electronics and computers. In each case, a core project is discussed and step-by-step instructions are given for its assembly and testing. After the core project is completed, suggestions are offered for project extensions. Expanding the project can include improving the electronic circuit, adding new features to the programs that support the project, or doing additional experiments.[1]

Most laboratory interfacing projects can be classified as either input devices (measurement with the computer) or output devices (controlling things with the computer). This book includes several of both types. There are projects to measure temperature, position, light intensity, pressure, magnetic fields, absorption of light by a solution, and even the speed of sound. There is a chapter on controlling stepper motors, which is a perfect start for a robotics project. Another lets you control a voltage output with the computer. Several of the projects are harder to classify. One lets you use your Apple II computer to run a TV-type game show. Another allows you to experiment with computer vision and artificial intelligence. One allows you to use your computer to check out and recharge batteries. Two of the projects used together allow you to digitally record sounds and then play them back. Two disks full of pre-recorded sounds are included with the book. You can have your computer say hello or play a bit of music. The title project introduces the important new science of chaos by studying the dripping of a faucet.

> All of the projects in this book were designed to be used with an Apple II+ (64k), Apple IIe or Apple IIGS. A few of the projects can also be done using the Apple IIc, IIc+ or Laser 128. Many cannot be done with these computers. Check the "Project Requirements" section of each chapter if you want to determine what model of Apple II can be used.

This book involves three academic areas: science, electronics, and computers. We feel strongly that all three subjects are best learned by experimentation. We encourage you to try these projects, add ideas of your own, and learn as you go. Using the Apple II, you can build laboratory interfacing projects inexpensively and with very little chance of damaging your computer. These projects were selected

---

[1]We would like to hear about interesting project extensions that you develop. We may mention your ideas in future revisions of this book or in *The Caliper*, the Vernier Software newsletter. A *Feedback Form* is provided in the back of this book.

because they are relatively simple to build and provide a useful finished product. We think you will find these projects to be educational and fun to build and use.

Each project is described in a chapter that consists of:

- **Project Requirements**: Information on the background knowledge and experience needed in electronics and programming, lists of parts, and supplies and tools needed for the project. This section also specifies which models of Apple II computers can be used.
- **Background Information**: A look at the electronics and the science behind the project.
- **Step-by-Step Instructions**: Detailed instructions on how to build and use the core project.
- **Troubleshooting**: Suggestions on what to do if the circuit does not work correctly.
- **Project Extensions**: Suggestions for extending the project by changing the hardware, improving the software, or doing additional experiments with the device.
- **Supplemental Information**: Additional science and math background information, data on the electronic components used, or explanations of how the programs operate.

The *Chaos in the Laboratory Programs* disks included with this book have sample programs that will let you try out your project with the computer. Most of these are short, straightforward BASIC programs. They include many comments (in REM statements) to explain how the programs operate.

Before you begin assembling anything, we recommend you read the entire *Introduction* of this book, which contains information that will be helpful in selecting parts and in assembling the projects. It also gives you an explanation of the various connections provided on the Apple II game ports.

When selecting a project to undertake, look over the "Project Requirements" at the beginning of each chapter. Choose a project that is within your abilities. Some projects require previous electronics assembly experience. Do not "bite off more than you can chew." Remember, you can always expand on a project if time allows. In fact, the core project explained in the "Step-by-Step Instructions" may be just the starting point for an impressive science project. The chart on the next page may also be used to help you decide on the right project. In the second column, each project has been assigned an arbitrary "difficulty factor." This number is based on discussions with our test assemblers. If you have never soldered before, start with one of the projects that has a low difficulty factor. After you do a few of these, you will be ready for a more complex project. The chart also indicates the approximate time that our test assemblers spent building each core project. The time listed does not include time spent doing project extensions.

Another factor to consider in selecting a project is the availability and cost of the necessary parts. Column four of the chart indicates whether or not it is likely that you will be able to buy the necessary parts at local electronic-parts suppliers. For several of the projects (marked in column five), Vernier Software sells parts kits.[2] For three of the projects, we have custom-printed circuit boards. The custom circuit boards greatly simplify project assembly. These projects are marked with a "+" sign on the chart. The last column is an estimate of the approximate cost of the necessary components. This is the cost assuming that you need to buy all of the parts new.

The chart also identifies projects that have prerequisites—projects that cannot be done until another project is completed. There are four projects that require a device to read a voltage into the computer.

---

[2]Details on the materials included in each parts kit are explained in the "Project Requirements" for the project. A Parts Kit Order Form is included at the back of this book.

Before you can use these projects you need either an A-to-D Converter (Project 5 in this book) or a Voltage Monitor (Project 7 in the book *How to Build a Better Mousetrap*).[3]

Project	Difficulty Factor (1 = Easy 10 = Hard)	Approx. Assembly Time (Hours)	Parts Available Locally?	Vernier Software Parts Kit Available?	Approx. Cost of Parts
1 - Thermistor Temperature Probe	2	2	Yes	Yes	$10.00
2 - Quiz Show	4	3	Yes	No	$20.00
3 - Computer Vision	4	4	Yes	No	$20.00
4 - Colorimeter	8	6	Yes	No	$12.00
5 - A-to-D Converter	3	2	No	Yes	$32.00
6 - Light Meter *	4	2	No	No	$16.00
7 - Barometer / Pressure Sensor *	3	2	No	Yes+	$42.00
8 - Magnetic Field Sensor *	4	3	No	Yes	$24.00
9 - Smart Battery Charger *	9	5	No	No	$19.00
10 - D-to-A / Digitized Sound †	6	3	No	Yes+	$21.00
11 - Stepper Motors	5	2	No	Yes	$16.00
12 - Speed of Sound	5	3	Yes	Yes+	$19.00
13 - Ultrasonic Range Finder	5	2	No	No#	$75.00
14 - Chaos in the Laboratory	5	2	Yes	Yes	$12.00

* Also requires A-to-D Converter (Project 5) or other voltage input device
† Requires A-to-D Converter (Project 5) for digitizing new sounds
\# Parts kit available from CAPE-tech
\+ These parts kits include custom circuit boards

> All of the circuits in this book have been used extensively at Vernier Software and at our test sites. It is unlikely that you would damage your computer in attempting to build or use any of the circuits in this manual. However, you should take care to double-check your wiring before you connect the circuit to the computer. Vernier Software shall have no liability or responsibility to any person or entity with respect to any liability, loss or damage caused or alleged to be caused during construction or use of any of these circuits.

NOTE TO TEACHERS

We hope this book can provide a start to a number of independent research or "science fair" projects. The core projects are explained in considerable detail and should be within the capabilities of most students. Each project has been assembled by several test assemblers, some of whom had very little experience in electronics. The "Project Requirements" section at the start of each project gives our estimate of the background experience needed for successful completion of the core project. Each project is left open-ended, with a number of different suggestions given for further work. Students who are interested in programming can improve the simple programs on the *Chaos in the Laboratory Programs* disks. Students who are interested in electronics can often make changes or additions to the circuitry. A third type of project extension is to do further experimentation using the project as a tool.

[3]These projects can also be done using a Voltage Input Unit from Vernier Software. See *Appendix F - Teacher's Guide* for more information on using these projects with other voltage input hardware and even other computers.

The student will have a chance to show some creativity, initiative, and thoroughness with these project extensions.

Appendix F of this book is a *Teacher's Guide*. It includes comments on the difficulty and complexity of the projects. It also includes results that we obtained in our tests of these projects and suggestions for how the completed projects can be used for teacher demonstrations.

The *Chaos in the Laboratory Programs* Disks

Programs to use with each of the projects are provided on the *Chaos in the Laboratory Programs* disks, which are attached to the inside back cover of this book. Both sides of both disks are full of programs and sample files. Many of the sample programs are short, simple Applesoft BASIC programs. In many cases there are missing features that any good program should have (such as rounding off of numbers or provisions for an easy way to quit). The programs are provided in this condition to encourage students to examine them and add new features.

Several of the programs on the disks are in a more finished form. In these cases, the minimum software required for the project was complex enough that most student programmers would not be able to work with it easily. These programs typically involve a combination of BASIC and machine-language programming. Examples include the digitized sound programs, one of the Ultrasonic Range Finder programs, and the oscilloscope program for use with the A-to-D Converter.

The *Chaos in the Laboratory Programs* disks use the ProDOS operating system. They are not copy protected. They are copyrighted, but you are permitted to make copies for your own use. Note that the ProDOS operating system requires 64k of RAM memory. All Apple IIe, IIc, IIc+, IIGS and Laser 128 computers have at least that much memory, but some Apple II+ computers may need to have memory added to reach the 64k minimum.

Before you do anything else, please make copies of all four sides of these disks and put the original disks in a safe place. If you have only 5.25 inch disk drives, copy the four sides onto four 5.25 disks. If you have access to 3.5 inch disks, we encourage you to copy the four sides onto one 3.5 inch disk in four subdirectories (it will all fit, but only if you use subdirectories).

Here is an overview of how the disks are set up:

Disk #1—Front Side: Volume name /CHAOS1
Contains all of the programs and files for use with these projects:
- Thermistor
- Quiz Show
- Computer Vision
- Colorimeter
- Speed of Sound
- Ultrasonic Range Finder
- Chaos in the Laboratory

+ Startup Program
+ Update file
+ ProDOS operating system (you can start your computer using this disk)

Disk #2—Front Side: Volume name /CHAOS2
Contains all of the programs and files for use with these projects:
- A-to-D Converter
- Light Meter
- Barometer/Pressure Sensor
- Magnetic Field Sensor
- Smart Battery Charger

• D-to-A Digitized Sound (all files except those related to sound)
• Stepper Motors
+ ProDOS operating system (you can start your computer using this disk)

Disk #1—Back: Volume name /S1
Contains the programs and some sound files for use in digitized sound experiments.
+ ProDOS operating system (you can start your computer using this disk)

Disk #2—Back: Volume name /S2
Contains only sound files for use with digitized sound experiments.

Appendix D is a complete list of the programs and files used with each project.

The *Chaos in the Laboratory Programs #1* disk also includes a program named UPDATE. This program will display all of the known corrections that should be made to this book. Run the UPDATE program as soon as you receive this book and correct any errors that are listed. If you find errors in the book that are not mentioned in the UPDATE program, please contact us. A *Feedback Form* is included in the back of this book.

If you are a novice at using a computer or the ProDOS operating system, you may find the following helpful:

- To start up one of the *Chaos in the Laboratory Programs* disks: Place the disk in the disk drive labeled #1 and turn on the computer.
- To see a list of all of the programs on the disk: Type CAT and then press the <RETURN> key.
- To start a particular program: Type RUN (program name) and then press <RETURN>.
- To stop a program: Press the <CONTROL> key followed by the <C> key, and then press <RETURN>.
- In many of the programs you are asked to enter values or make choices. Often, there is a value displayed inside [ ] symbols. This is the default choice. If you simply press <RETURN>, this value will be taken as your choice.
- To look at all of the lines of a program: Quit the program, type LIST and then press <RETURN>.
- To look at a particular line of a program: Quit the program, type LIST followed by the line number and then press <RETURN>.
- To change a particular line of a program: Quit the program, type the line number followed by the entire (changed) line and then press <RETURN>.
- When you change from one program to another, the best way to assure that everything will work correctly is to reboot the computer.

For people with computer experience only:

- These programs can be easily transferred to a 3.5 inch floppy disk. All four sides will fit on one 3.5 inch disk with room to spare. You must set up subdirectories on the disk. The extra speed and room for data storage will make the programs easier to use.
- These programs can also be transferred to a hard disk, if you have one available. You must set up subdirectories on the hard disk. This will greatly reduce disk access times for the programs.
- The reason you have to set up subdirectories when copying the *Chaos in the Laboratory Programs* disks onto 3.5 inch disks or hard disks is that there are too many separate files to fit in one ProDOS disk or subdirectory.

- Many of the programs load machine-language files as they start up. To operate properly, the PREFIX must be set properly so the program can find the files. If you are using floppy disks, the easiest way to do this is to simply reboot, using the appropriate disk before you start. If you do not want to reboot, just use the PREFIX command to set the prefix before running the program.

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

ProDOS is a copyrighted program of Apple Computer, Inc. licensed to Vernier Software to distribute for use only in combination with the *Chaos in the Laboratory Programs* disks. Apple software shall not be copied onto another disk (except for archive purposes) or into memory unless as part of the execution of programs on the *Chaos in the Laboratory Programs* disks. When the programs have completed execution, Apple software shall not be used by any other program.

Electronic Components

This chapter contains a quick look at the "building blocks" of the electronic circuits described in this book. It provides background information that should prove helpful when selecting and using components. It also reviews the symbols used in schematic diagrams. Obviously, this chapter cannot cover all aspects of electronics. The information is designed to help get you ready to do the projects in this book. For more information on electronics, you may wish to refer to some of the recommended texts listed in *Appendix C*.

Before we begin this discussion, we should explain the use of two terms that sometimes cause confusion. The term *current* is used in this book to mean the flow of **positive** charge. Science texts sometimes use a different definition of the term current. The term *voltage* is used instead of the more technically correct *electrical potential difference*. Unless otherwise specified, voltage is measured relative to ground. With both *current* and *voltage*, we are using the standard terminology used in electronics.

Resistors

Resistors are devices that limit the current in a circuit. There are two major categories of resistors: *fixed-value resistors* and *variable resistors*.

FIXED-VALUE RESISTORS

Almost every project in this book uses fixed-value resistors. These resistors have a specific resistance value in units of ohms. The symbol for ohms is the Greek letter capital omega: Ω. The term kilohm (kΩ) is used for a thousand ohms, and Megohm (MΩ) is used for a million ohms. The value of a fixed resistor is most often labeled on the resistor by a color code. This color code is explained in *Appendix B*.

On schematic diagrams, the symbol for a fixed resistor is:

The resistance value (10 kΩ in the diagram above) is indicated next to the resistor symbol. Another specification often quoted for resistors is their *power rating*. This is a measure of how much power they can handle without overheating. In general, the physical size of the resistor increases as the power rating increases. One-fourth (1/4) watt resistors are large enough for most of the projects in this book. When a resistor with a larger power rating is required, we will specifically mention it in the instructions.

There are actually two different types of fixed resistors that we will use in projects in this book:

Carbon-Composition Resistors

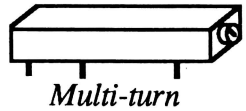
These are the standard, inexpensive resistors that are seen everywhere. They are usually labeled with four colored bands that identify their resistance value and their tolerance (the amount by which they may differ from their labeled value). See *Appendix B* for the resistor color code. In most cases, these resistors are guaranteed to be within 5% of their labeled value.

Metal-Film Resistors

These resistors are slightly more expensive, more accurate, and more stable. They are usually labeled with five color bands to indicate their resistance value and their tolerance. In most cases, metal-film resistors are guaranteed to be within 1% of their labeled value. These resistors also maintain their resistance value very well when their temperature changes. This type of resistor will be recommended when the resistance value must be controlled very accurately.

VARIABLE RESISTORS

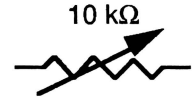
Many projects require a resistance that can be easily changed. A variable resistor has three terminals. Two of the terminals are at the opposite ends of the device. Between the two end terminals is a *slider,* which can move to different positions. Two types of variable resistors are illustrated below.



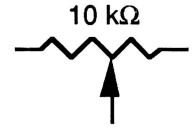
Multi-turn Variable Resistor

Single-Turn Variable Resistor

In most applications, one of the end terminals and the slider terminal are used. The resistance between these two terminals varies as the position of the slider changes. A variable resistor connected in this way is sometimes called a *rheostat.* A typical use for a rheostat would be to control the amount of current flowing in a part of a circuit. The schematic diagram symbol for a rheostat is also shown at the right. The number labeled indicates the maximum resistance.

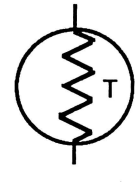


When all three terminals of a variable resistor are connected, it acts as a voltage divider, supplying a controlled fraction of the voltage to the slider. A variable resistor connected in this way is called a *potentiometer.* Potential is another term for voltage, so a potentiometer is a device that controls voltage. The symbol for a potentiometer is shown at the right. The arrow indicates the slider terminal.



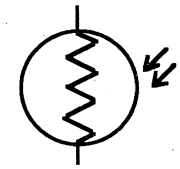
Thermistors

Thermistors are resistors that vary with temperature. Most thermistors have negative temperature coefficients; that is, their resistance decreases as the temperature increases.



Photoresistors

Photoresistors are resistors that vary as the intensity of light striking them varies. Photoresistors are sometimes called photocells (for example, in Radio Shack catalogs).[4] Most photoresistors are made using cadmium-sulfide.

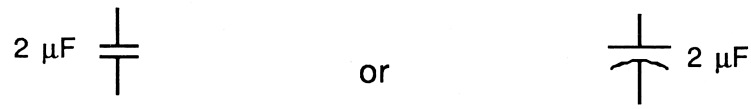


Capacitors

Capacitors are devices that store electrical charge. They are available in many different types, including electrolytic, tantalum electrolytic, ceramic, polyester, mica, and polycarbonate. Each of these capacitor types differs in materials used and in construction. They also differ in cost, capacitance value, stability, and physical size. One important difference is that some capacitors are *polarized.* Polarized capacitors have labeled positive and negative leads. When placing polarized capacitors in circuits, be sure to orient them in the correct way. Other capacitors are nonpolarized and may be oriented either way.

The two most important specifications about a capacitor are its *capacitance* and its *voltage rating*. Capacitance is a measure of how much charge a capacitor can store per volt. It is measured in a unit called a *farad*.[5] Most capacitors have values that are much smaller than a farad, so the most commonly used units for capacitors are microfarads (μF), which are equal to 10^{-6} farads, and picofarads (pf), which are equal to 10^{-12} farads. The other important specification for a capacitor is its voltage rating, which is the maximum voltage that the capacitor can handle without breaking down. When selecting a capacitor, make sure that it has at least the voltage rating required. It does not hurt to use a capacitor with a higher voltage rating.

The schematic diagram symbol for a capacitor is shown below. If the capacitor is polarized, a "+" symbol is used to indicate the positive lead.



Diodes

Diodes act as "one-way valves" for electrical current. The two most important specifications used with diodes are the *current rating* and the *peak inverse voltage rating*. The current rating is the largest current that can safely go through the diode (in the direction the diode allows current to go). The peak inverse voltage rating (PIV) is the largest voltage that the diode can have across it without breaking down and conducting in the reverse direction.

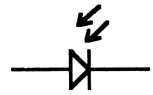
Diodes are typically marked with a band on one end marking the negative end. The schematic diagram symbol for a diode looks like an arrow pointing in the direction that **positive** current flows. The diagram below shows a typical diode and its corresponding schematic symbol.

[4]The term *photocell* should really be used only for devices that convert solar energy into electricity. Photoresistors do not do this—they simply control a current by changing resistance.

[5]A capacitor has a capacitance of one farad if it stores one coulomb of charge when the voltage across it is one volt.

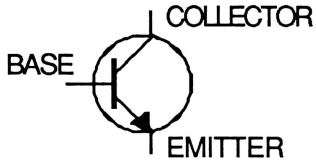
Photodiodes

Photodiodes are similar to ordinary diodes except that when light strikes the junction between the two types of semiconductor, electrons can move though the photodiode. This effect is typically used to measure light levels. Photodiodes come in a variety of packages. The schematic symbol for a photodiode is the standard diode symbol with light striking it.

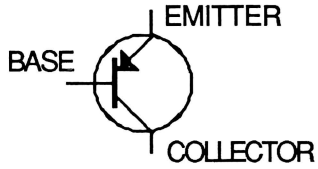


Transistors

There are hundreds of different types of transistors. They are used as amplifiers, in oscillators, in power supplies, and in almost any complicated electronic circuit. For the projects in this book, the main use of transistors is as electronic switches. The D-to-A Converter project uses a transistor as a current booster, allowing you to control a larger current with signals from the computer. The transistor conducts electricity between its *collector* and *emitter* terminals whenever current enters (or leaves) its *base* terminal. The types of transistors used in projects in this book are:



Bipolar NPN Transistor



Bipolar PNP Transistor

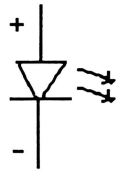
A little trick to use to keep these two transistor symbols straight is: NPN stands for **N**ot **P**ointing i**N**, and PNP stands for **P**ointing i**N**, both referring to the direction of the arrow on the diagram. Notice that the arrow always indicates the emitter of the transistor.

Identifying the leads on a transistor can be a little tricky. We will include diagrams in the "Step-by-Step Instructions" showing how to identify the base, emitter and collector leads on the transistors required.

LEDs

LED stands for *Light-Emitting Diode*. They are just that—diodes that give off light. The light emitted can be red, green, yellow, or infrared. LEDs should always be oriented properly. The negative lead of the LED is typically marked by a flat spot on the plastic case. You must always have a resistor in series with the LED to limit the current flowing through it. The symbol for an LED is the same as the symbol for a normal diode, with small arrows indicating the light given off.

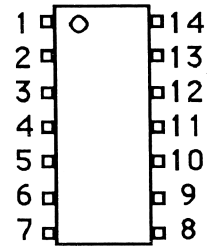
ACTUAL LED



SCHEMATIC SYMBOL

Integrated Circuits

Integrated circuits (ICs) are complex, miniaturized circuits designed for a particular task and sold as a single unit. ICs can contain dozens (or even hundreds) of resistors, capacitors, diodes, and transistors. Most of the integrated circuits used in these projects are packaged in *DIP* packages. DIP stands for "**D**ual **I**n-line **P**ins," indicating that the connection pins are in two parallel rows. For all ICs of this type, the pins are numbered in a counterclockwise direction as seen from the top (with the pins pointing away from you). The #1 pin is marked with a dot nearby or sometimes with an indentation in the case. A top view of a typical 14-pin DIP IC, with the pins numbered, is shown below:



Typical DIP IC (Top View)

When working with integrated circuits, be very careful about identifying the pins. Remember that the illustration above is a top view. If you are soldering on the underside of the IC, the pins appear numbered in a clockwise fashion.

When DIP integrated circuits are needed for these projects, IC sockets will always be recommended. The socket is soldered into the circuit, and the IC is plugged into the socket after all of the soldering on the circuit board is completed. A socket provides two advantages: (1) It allows you to easily change ICs if there is ever a need to do so; and, (2) The IC is kept away from the heat of the soldering iron. When you place the IC in the socket, be very careful to place pin #1 of the IC into the proper hole. Also, avoid bending and breaking the thin pins of the IC.

The schematic diagram of an IC depends on the internal circuitry contained in the IC. ICs are used as building blocks in modern electronics, and often it is not necessary to understand how the IC operates internally as long as you know what the IC does. Several different types of integrated circuits are used in the projects in this book. In each project where an integrated circuit is used, its function, pin connections, and wiring information will be given.

Switches

Three major types of switches are indicated symbolically as indicated below.

Rotary Switch

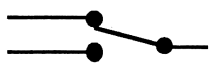
Push-Button Switch

Toggle Switch

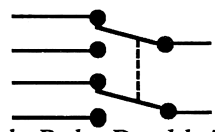
A rotary switch allows contact to be made between one terminal and any of a number of other terminals.

A push-button switch can be made in several different ways. Most push-button switches are normally open. This means the switch does not conduct, except when the button is pushed. Other push-button switches are normally closed. Some make momentary contact briefly when they are pushed. Others change from conducting to open or vice versa each time they are pushed.

Toggle switches come in many varieties. They are described by the number of *poles* they have and the number of *throws*. The number of poles describes how many separate electrical circuits can be wired to the switch. If a switch has two poles, it is like two switches in one, controlled by the same lever. The number of throws indicates how many different contact points there are for each part of the switch. The diagrams below indicate two different toggle switch configurations.



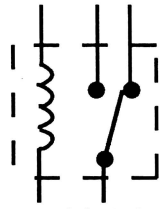
Single Pole, Double Throw



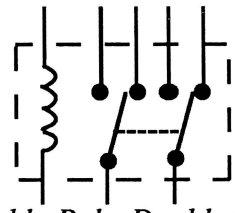
Double Pole, Double Throw

Relays

Relays are electrically-operated switches. The relay includes an electromagnet (often called the *coil*) which, when activated by a current flow, causes a switch to change positions. Relays, like switches, can be single or double pole, and single or double throw. Relays are also described by the voltage at which the coil is designed to operate and by the amount of current that can be safely passed through the relay's contacts.



Single-Pole, Double-Throw Relay



Double-Pole, Double-Throw Relay

Wire

Several terms are used to describe wire in these projects:

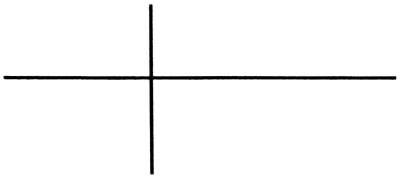
x-conductor cable: The "x" refers to the number of separate wires that are packaged together.

shielded cable: This consists of wire(s) that are contained inside a "tube" usually made of braided, copper wire.

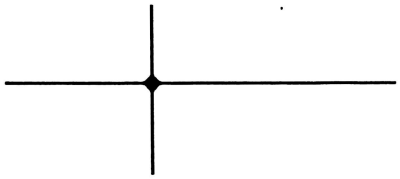
coaxial cable: A cable with a center strand of wire (or wires) completely surrounded by an outer conductor. The outer conductor is usually connected to ground, and it shields the inner wire from electrical interference.

stranded wire: Stranded wire is made of many fine, metal wires twisted or braided together. Stranded wire is much more flexible than a solid wire of the same size and is recommended in any situation where the wire will be bending during use.

The diagrams below show the schematic symbols for wires that are crossing with and without electrical connection between them. These symbols can sometimes cause confusion if not understood.



TWO WIRES CROSSING
NOT CONNECTED



TWO WIRES CROSSING
CONNECTED

Sources of Electronic Components

This section gives suggestions on where to purchase electronic components to build the projects in this book. Electronic-parts suppliers frequently change part numbers and discontinue parts. At the time of publication, all references to part numbers were correct. We are sure this will not be true for long. The program UPDATE on the *Chaos in the Laboratory Programs #1* disk will display any known changes in part numbers or availability at the time we ship this book to you.

Mail-Order Parts Suppliers

Before assembling any of the projects in this book, we recommend writing for free catalogs from a number of mail-order, electronic-parts suppliers. These companies have a wide variety of parts and

usually offer low prices. Even if you do not purchase your components via mail, their catalogs will provide a good source of information about what is available, and they will give you an idea of what you should expect to pay if you purchase the components locally. A list of a few parts houses is provided below:

Mouser Electronics
11433 Woodside Ave
Santee, CA 92071
(800) 346-6873
or
Mouser Electronics
2401 Hwy 287 North
Mansfield, TX 76063
(817) 483-4422

All Electronics
14004 Oxnard St.
P.O. Box 567
Van Nuys, CA 91408
(800) 826-5432

Marshall Industries
9320 Telstar
Elmonte, CA 91731
(919) 878-9882 or
(800) 522-0084

Digi-Key Corporation
P.O. Box 677
Thief River Falls, MN 56701
(800) 344-4539

Jameco Electronics
1355 Shoreway Road
Belmont, CA 94002
(415) 592-8097

Newark Electronics
4801 N. Ravenswood Ave.
Chicago, IL 60640
(800) 367-3573

R&D Electronics
1224 Prospect Ave.
Cleveland, OH 44115
(216) 621-1121

Radio Shack Stores

Radio Shack stores are located in almost every community. They provide a convenient source for many components. In some cases Radio Shack sells components that are not available from other sources. We list Radio Shack part numbers for many of the parts used for these projects. Unfortunately, Radio Shack often changes part numbers or discontinues products.

Local Electronic Parts Suppliers

If you live in a large town, check your telephone book for "Electronic Equipment and Supplies." You may find a local dealer who can supply components at very good prices. On the other hand, be aware that some of these businesses will not want to deal with small purchases. They may have large minimum orders, or they may not have very many components in stock. Hamilton-Avnet and Newark Electronics are two large parts distributors with offices in most states. They are sometimes willing to sell small quantities of components.

Vernier Software Parts Kits

Vernier Software sells parts kits for some of the projects. Each parts kit contains all the electrical components for the core project. For a few projects, we sell two versions of the parts kit: One has a 16-pin ribbon cable for connection to the internal game port on an Apple II+, IIe, or IIGS. The other version has a 9-pin connector for connection to the external game port. The portion of the order code following the hyphen indicates the type of cable included.

For three of the projects (Barometer / Pressure Sensor, D-to-A / Digitized Sound, and Speed of Sound) the Vernier Software parts kit has an additional advantage. In these cases a customized circuit board is included, which makes the project assembly much easier.

The parts kits do not include tools, supplies or solder. With the exception of the A-to-D Converter, these kits do not include boxes for housing the circuit. The following parts kits are now available:

Project 1.	Thermistor Temperature Probe (CTP-9 or CTP-16)	$10.00
Project 5.	A-to-D Converter (CAD-16)	$32.00
Project 6.	Light Meter (CLM-DIN)	$16.00
Project 7.	Barometer / Pressure Sensor (CBR-DIN)	$42.00
Project 8.	Magnetic Field Sensor (CMG-DIN)	$24.00
Project 10.	D-to-A / Digitized Sound (CDA-16)	$21.00
Project 11.	Stepper Motors - Unipolar (CSU-16)	$16.00
Project 12.	Speed of Sound (CSS-9 or CSS-16)	$19.00
Project 14	Chaos in the Laboratory (CCH-9 or CCH-16)	$12.00

Prices and availability are subject to change. Include $4.00 per order for shipping. An order form can be found at the back of this book. These parts kits can be ordered from:

Vernier Software
2920 S.W. 89th. Ave.
Portland, OR 97225
(503) 297-5317

In addition, the Ultrasonic Range Finder (Project 13) is available in a kit from CAPE-tech, Box 115, Wynnewood, PA 19096-0115, phone (215)-789-0924.

How to Solder

Each of the projects in this book involves assembling an electronic circuit. In each case the "Step-by-Step Instructions" suggest soldering the circuit together. We recommend soldering because it produces a finished product that you can use for some time to come. There are some alternatives to soldering which are discussed briefly at the end of this section.

Soldering is not difficult. If you have the right equipment you can learn to make good solder joints in a few minutes. The basic idea is to heat the objects to be soldered, encase them in solder in such a way that the solder actually intermingles with the metal of the objects, and hold them stationary while the solder cools.

Equipment Needed

SOLDERING IRON

A good soldering iron makes soldering easy; a bad one can mean that no matter what you do you will not get good solder joints. Use a "pencil-type" soldering iron. Soldering irons with temperature-regulated tips are the best (but expensive). If a temperature-regulated iron is not available, use a 25- to 35-watt iron with a fairly narrow tip. Recommended Radio Shack models are #64-2070 and #64-2067. Do not use a large, high-wattage soldering iron or gun for these projects. These irons will overheat and damage components and circuit boards.

Many soldering irons have replaceable tips. Check the tip of the iron you are using. If it is worn down and corroded, consider replacing it. Soldering iron tips are inexpensive.

Keeping the tip of the iron clean makes a tremendous difference in the rate of heat transfer. Ironically, the cleaner and hotter the tip, the **less** likely that you will damage components while soldering. If the tip is clean and hot, you can very quickly complete the soldering operation before much heat travels to the sensitive part of the component. If the tip is dirty, heat transfer is very slow, and you will have to hold the iron on the joint for a long time to get it hot enough to melt the solder. By this time, considerable heat has flowed into the component.

SOLDER

Use narrow-gauge, rosin-core solder. Thin solder melts quicker and is easier to control. Recommended Radio Shack solder rolls are #64-005 and #64-006. Solder with a diameter larger than 0.050 inches is difficult to use when assembling circuit boards. **Never** use acid-core solder. The acid will cause corrosion of metal wire and other components. Solder is a mixture of tin and lead. Solder rolls are typically labeled with two numbers, indicating the percentages of tin and lead. 60/40 solder is generally considered the best for hand-soldered projects. The reason for this is that a 60/40 mixture of tin and lead actually melts at a lower temperature than either tin (450°C) or lead (621°C) alone. 60/40 solder melts at about 361°C.

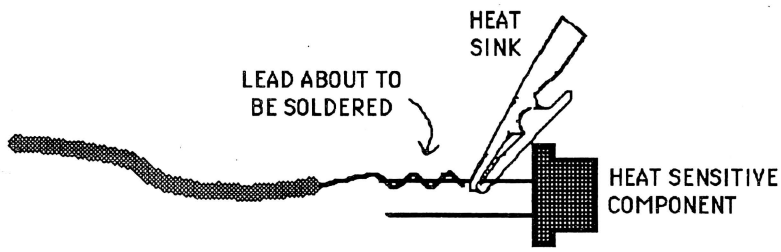
The rosin inside the solder acts as a *flux*. A flux is a chemical which, when heated, helps remove oxidation from the surface of metals. All solder used in electronics work should be made in the form of a tube (or multiple tubes) with rosin filling the tube. This rosin is automatically applied to the metal each time you solder.

SPONGE

Keep the tip of the iron clean by periodically wiping it on a damp sponge. The tip of the soldering iron should always look bright and shiny. If you do not have a sponge handy, you can use a rag moistened with water.

HEAT SINK

When soldering any component that can be damaged by overheating (in particular, transistors, LEDs and voltage regulators), use a *heat sink*. A pair of pliers, an alligator clip, or any metal object can serve as a heat sink. The idea is to provide a large metal object into which heat can flow so that the heating of the component is reduced. Place the heat sink between the solder and the component, as shown in the figure below:



A Heat Sink in Use

OTHER SOLDERING AIDS

If you plan to do a lot of soldering, you may want to get a device to hold the components as you solder. "Helping Hands Project Holders" are sold by Radio Shack (#64-2093) and other electronics equipment suppliers. This is a small stand with two alligator clips that can be used to hold wires, circuit boards, and components. A small vise can also be used for this purpose.

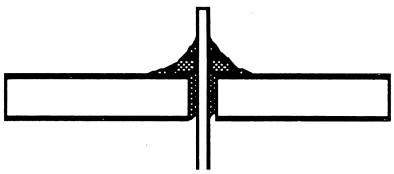
The Soldering Procedure

1. Make a good physical contact between the parts to be assembled (by twisting wires, etc.) before you start to solder.
2. If the objects to be soldered are oily, greasy, rusty, or coated in some other way, clean them with alcohol or by sanding with fine sandpaper.
3. Wipe the tip of the soldering iron on the dampened sponge each time before you use it. The soldering iron tip gets coated with dirt and burned flux as you use it. Wiping off this material makes the transfer of heat from the iron to the metal more efficient.
4. Touch the tip of the iron to the solder. This will cause a small amount of solder to melt and coat the tip. This is called *tinning*. Tinning prevents oxidation of the tip and also makes heat transfer

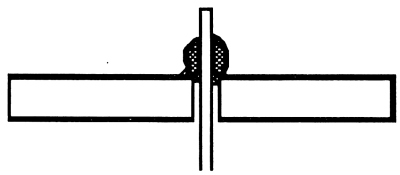
more efficient. Tinning will not normally be needed except when you first heat up the soldering iron.

5. Start heating the connections to be soldered. Always remember the old saying, "**Heat the metal parts, not the solder.**" Place the heated tip of the iron against the parts to be soldered. Try to maximize the contact area between the tip and the metal. Make sure that the tip of the iron is heating **both** of the objects being soldered. Be patient!
6. After a few seconds, move the solder in and let the metal heat up the solder. Wait until the solder flows and looks like mercury. Apply just enough solder to coat the objects being soldered. If you are soldering to a large piece of metal, give the iron a little extra time to heat the whole piece of metal.
7. Move the iron away and hold everything steady for a few seconds. If the connections move during the cooling, the solder may not solidify properly. A *cold solder joint* may be the result. This is a soldered connection that does not make good electrical contact. It may appear dull (instead of shiny) and rough (instead of smooth).
8. Inspect the connection. It should be bright and shiny. If it looks rough and dark, it is not a good connection. There may not even be an electrical connection between the parts. Reheat the joint and try again. Check the connection by trying to pull the objects apart. If you feel any looseness, it is probably not a good electrical connection.

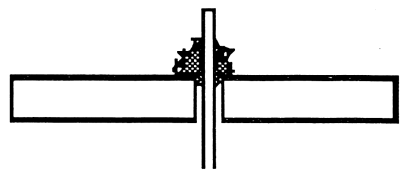
The diagrams below illustrate what good and bad solder joints on a circuit board look like. Note the shape of the solder on a good solder joint.



good solder joint
(smooth and shiny)



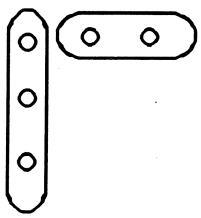
bad solder joint
(solder did not flow)



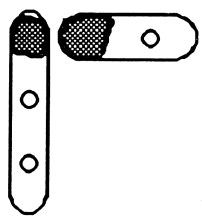
bad solder joint
(dark, rough and crumbly)

FOLLOW-UP PROCEDURE

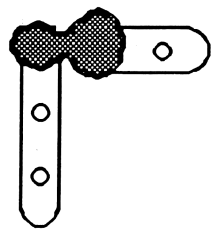
- Check for *solder bridges*. Sometimes when you solder on a circuit board, solder can connect across places that should not be in electrical contact. If necessary, reheat the connection and remove the extra solder. In some cases, you may need to use an unsoldering tool (see below).



circuit board
before soldering



properly soldered



solder bridge

- Clip off excess lead wires near the solder junction. If these leads are not cut short enough, they can bend over and create a short.
- (Optional) If the appearance of the project is a high priority to you, you may want to use a rag or a toothbrush dipped in alcohol to remove the excess rosin from the project. This rosin will not hurt anything, but it can sometimes cause discoloration in time. One advantage of removing the rosin is that it makes solder bridges easier to spot.

Unsoldering

Unfortunately, any time you solder together a project, there is a chance that you may have to unsolder a connection or two. Unsoldering is not difficult. In most cases you will want to remove most of the solder used in the original connection. There are two ways to do this. Either method requires an inexpensive tool.

- Using soldering braid: This is a fine woven mesh of copper wire (for example, Radio Shack #64-2090). It is placed on top of the connection to be unsoldered and then the tip of the soldering iron is placed on it. The copper braid soaks up the solder as soon as it becomes liquid. If you have not used solder braid, you will probably be surprised at how well it works.
- Using vacuum solder removers: There are several types of devices available which create a vacuum to suck up the liquid solder and help you unsolder. The least expensive of these are desoldering bulbs (Radio Shack #64-2086). Other "solder suckers" are also available (for example, Radio Shack #64-2120 and #64-2098). To use any of these devices, heat the solder until it becomes liquid, and then quickly move in the vacuum device to suck in the solder. Do not reuse the solder.

Alternatives to Soldering

If you want to try out the circuit without soldering it together, you can assemble the circuit on a solderless "breadboard" (for example, Radio Shack #276-175, #276-174, or #276-169). Using a breadboard is a good idea if you plan to make major changes in the circuit at a later time or if you do not plan to use the assembled circuit for very long. The disadvantages of prototyping boards are that they are expensive, and the poor connections you sometimes get can cause frustrations. Some high frequency circuits do not work well when built on prototyping boards due, to the size and shape of the internal connectors.

Another way to assemble the projects in this book is to use wire wrapping. Special wire-wrap tools are available that allow you to make all of the point-to-point connections you need. Special IC sockets are also used. If you already know the technique and have the tools for wire wrapping, you may want to build the projects in this book in that way. Otherwise, it is probably not worth your while to start wire wrapping on these projects.

Other Project Assembly Tips

- Avoid using solid wire, except for short jumper wires on a circuit board. Instead, use stranded wire (made up of many thin wires, twisted together). Solid wire will break if it is bent back and forth very many times. Stranded wire is more flexible and can withstand bending much better. When using stranded wire, however, watch for stray strands that can cause a short.
- Minimize wire movements in the final product. When wire leads are allowed to move back and forth, there are usually problems. This is especially true when a wire is allowed to bend at the point where it is soldered. Fasten wires down by tying them to something solid or by using clamps or glue.
- Put your assembled circuit in a plastic or metal box to protect it. This also minimizes wire movement.
- Be careful about static electricity when handling ICs. Touch a grounded metal object before touching the chip.
- When working with a circuit board, it is sometimes helpful to hold the board up to a light so that you can look at the non-plated side and see the plating pattern on the other side.
- Make your wiring neat. Keep leads as short as possible. Unnecessary loops of wire can cause problems by acting as antennae for electrical and magnetic fields. When placing electronic components on a circuit board, position them close to the circuit board. Neatness counts, even in electronics.
- Keep exposed, bare wire to a minimum. Even if the uninsulated parts of a wire are not touching anything, they may cause problems later when they are moved. If the circuit gets dropped or jolted,

the wire may accidentally touch something it should not touch. Keep leads short and use plastic tape or heat-shrink tubing as an insulator.

- When you think the project is complete and correct, take a few minutes to double-check the circuit before applying the power.
- For the first test of a circuit, briefly turn on the power and look, smell, and feel (carefully) for any overheating. If any component is hot, turn off the power. There must be something wrong.
- Heat-shrink tubing is very useful for insulating the connection between two wires. Heat-shrink tubing is plastic tubing that shrinks to about half its original diameter when heated. Slip the tubing over one of the wires to be soldered. After the soldering is completed, move it over the solder joint. Heat the tubing with the soldering iron or a hot-air gun, and it neatly shrinks down over the solder joint.

Troubleshooting

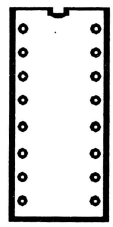
Troubleshooting is the process of trying to find and correct the problem when a circuit does not work. Anyone who builds electronics projects will need to do some troubleshooting. When you are faced with a circuit that does not work, your attack of the problem should go something like this:

1. Check for some of the easy-to-make mistakes, such as:
   - If you are using an integrated circuit, is the chip inserted the wrong way or are some of the pins missing the socket? A useful trick is to look carefully down the rows of pins on each side of the DIP IC. Pins that are missing the socket or bent will usually show up in this way.
   - Are the leads from the Apple game port used properly? If you are using a 16-pin connector to the internal game port, are you sure you have the plug inserted with pin #1 toward the keyboard?
   - The Apple IIe and IIGS both have two game ports. Make sure you do not have another device plugged into the other game port.
   - Is everything that is supposed to be soldered really soldered?
   - If you are using a circuit board, are there any solder bridges connecting areas that should not be connected?
   - If you are using any components that have identified leads (such as polarized capacitors, transistors, LEDs, or diodes), are the components oriented in the correct way?
2. If you still have not found an error, force yourself to go over the circuit again, wire by wire. If you have worked with the circuit for a long time, you may be too familiar with it. It is similar to proofreading your own writing after you have worked on it all day. Take a break and come back to it later or get a friend to look over the circuit.
3. If you still have not found a problem, resort to *scientific troubleshooting*. Scientific troubleshooting is based on the fact that if the circuit does not work there must be a logical reason. The steps of scientific troubleshooting are described below:
   - Study exactly how the circuit is acting. Does it do anything? Does part of it work? Is anything overheating? Is power being delivered to the circuit? Are the voltages what you would expect at key spots on the circuit?
   - What are the most likely things that could cause it to act like it is? Propose a best guess (hypothesis) about what could cause the circuit to this way. In troubleshooting, as in science, the simplest explanation is the most likely. In most cases, the problem is caused by something such as a wire in the wrong place, a bad solder joint, or an IC inserted backwards.
   - Test your hypothesis. Examine the part of the circuit that you suspect. If necessary, use a meter to check connections, voltages, etc.
   - If your first guess as to the problem is not correct, think about the situation again and come up with another hypothesis.
   - Keep proposing and testing ideas until you zero in on the real cause of the trouble. Only after you have considered all the simple things should you move on to more complex (and less likely) troublemakers such as bad ICs, cold solder joints, or errors in schematic diagrams.
4. Refer to the specific troubleshooting suggestions given for each project.

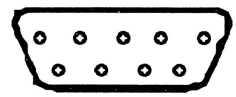
5. A *multimeter* is an invaluable tool for troubleshooting electronic circuits. A multimeter is a single meter that can be used as an ohmmeter (to measure resistance), a voltmeter (to measure electrical potential differences), and an ammeter (to measure currents). All three of these functions can be helpful in troubleshooting. We recommend that you obtain a digital multimeter, if possible. These meters have extremely *high impedance*, which means they do not draw much current from the circuit they are testing. This allows them to be used in places where a conventional meter will not work. There are several troubleshooting tests mentioned in this book that are only possible with a high-impedance meter.
6. If you are thwarted in all attempts to find the problem, try reheating every solder joint. Sometimes a connection that appears to be soldered really is not. By reheating the connection, you may fix a "cold-solder joint" that you did not even know about.
7. Double-check your wiring by using an ohmmeter. Make sure the points that are supposed to be wired together really are. The resistance between them should be nearly zero. These tests should be done with the circuit disconnected from the computer. They can help you locate incorrect wiring and identify solder joints that appear to be connected, but are not.

The Apple II Game Port

One of the things that has made the Apple II family of computers so popular is the game port. The game port provides several different ways of getting information into and out of the computer. Actually there are two different game ports that are used on Apple II computers. The Apple II+ computer has an internal 16-pin DIP game port. The Apple IIe and IIGS have this same internal game port, but in addition they have a second, external game port. This second game port has only nine connections on a sub-D connector.[6] Both game ports are shown below:



APPLE GAME PORT SOCKET (INSIDE II+, IIe, or IIGS)



GAME PORT SOCKET (OUTSIDE BACK OF IIe or IIGS)

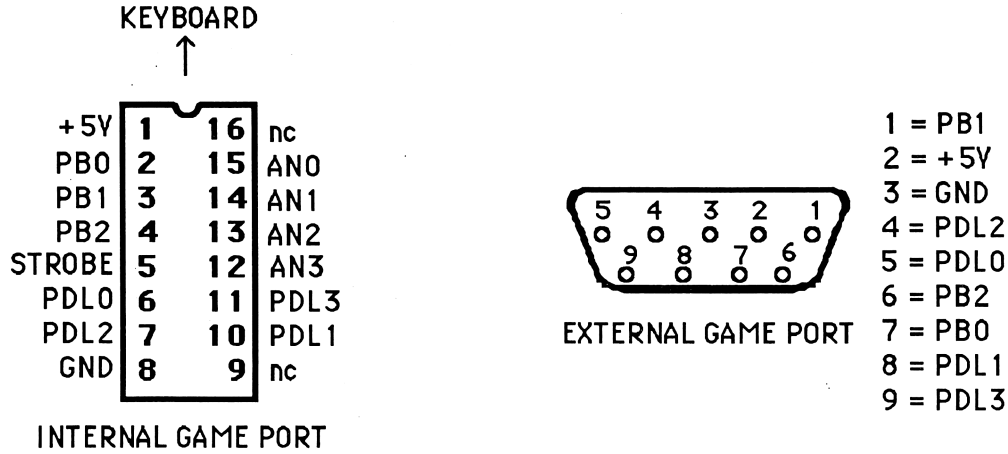
The Apple II Game Ports

The input and output connections of the game ports fall into six different categories:

- power-supply lead (+5V)
- ground lead (GND)
- switch inputs (PB0, PB1, PB2)
- paddle inputs (PDL0, PDL1, PDL2, PDL3)
- annunciator outputs (AN0, AN1, AN2, AN3)
- strobe output (STROBE)

Only the first four types of connections are available on the external 9-pin game port. The diagrams below show the pin connections for both game ports. *Appendix A* discusses cables that can be used to connect to the two Apple II game ports. Each of these game-port connections will be discussed below.

[6]The name comes from the "D" shape of the connector. The original D connector had 25 pins; those with fewer pins are called *sub-D*.



The Connections on the Game Ports

A Note about Apple IIc, IIc+ and Laser 128 computers

The Apple IIc, IIc+ and Laser 128 computers have only one game port. It is a 9-pin game connector similar to the 9-pin game port on the IIe and IIGS. These computers, however, lack the PDL2, PDL3 and PB2 lines. The PB0 and PB1 input lines on these computers are also a little different electrically from the switch input lines on the other Apple II computers. Circuits that work with the IIe switch inputs will not always work using a IIc. In addition, since these computers do not have a 16-pin game port, some of the connections that are available on the II+, IIe and IIGS computer are simply not available. As a result, about half of the projects in this book cannot be done using these three computers. At the beginning of each project there is a section called "Project Requirements" which will tell you whether a IIc, IIc+ or Laser 128 computer can be used for the project.

Power-Supply Lead (+5V)

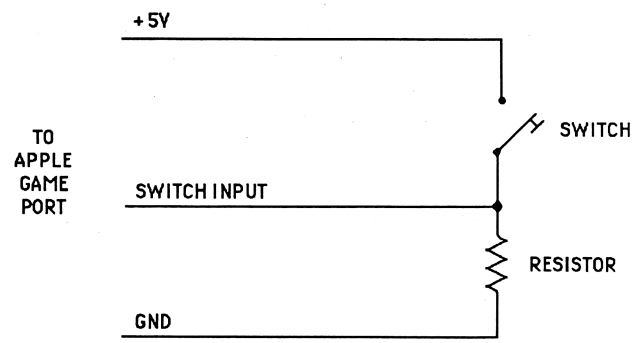
Either of the game ports has a lead labeled +5V. This lead is connected to the +5 volt output of the Apple II power supply. It can provide reference voltage levels for use in switches, or it can supply a small current for use in powering electronic devices. **The current drawn from the +5V line should never exceed 100 milliamperes.**

Ground Lead (GND)

Both the internal and external game ports have a lead labeled GND, which is electrically connected to the ground of the Apple II's power supply.

Switch Inputs (PB0, PB1, PB2)

There are three switch input connections available on either the internal or external game ports of an Apple II+, IIe or IIGS. The lines are labeled PB0, PB1 and PB2. They are *digital input lines*. This means that the computer can sense which of two possible conditions (low voltage or high voltage) is present on the line. The switch input lines are usually connected to the push-button switches on a game controller or joystick in a circuit similar to this:



In the diagram above, the voltage on the switch input line would be low if the switch were open, because the line is connected to ground through the resistor. If the switch is closed, the switch input line is connected to the +5 volt line, and the input voltage rises.

The computer can sense whether the voltage at the switch input line is low or high. A particular memory location is changed automatically as the voltage level at the switch input changes. The memory locations used are:

Switch Input	Memory Location[7]
PB0	49249 ($C061)
PB1	49250 ($C062)
PB2	49251 ($C063)

The number in each of these memory locations depends on the voltage at each of the three switch input lines. If the voltage at the switch input line is higher than 2 volts, the corresponding memory location will contain a number larger than 127. If the voltage on a switch input line is less than 0.8 volts, then the value stored in the corresponding memory location will be less than 128. If the voltage on the switch input line is in the range 0.8 to 2.0 volts, it may be interpreted either way.[8] One interesting thing about the way the switch inputs work is that the exact number stored in the switch input memory location is random. The voltage on the switch input line controls only whether or not the number is greater than 127.

A PEEK command in Applesoft BASIC can be used to check the status of a switch input line. Write a PEEK statement to determine the number stored in the memory location corresponding to the switch input line. Check whether or not this value is greater than 127. The statements below are examples:[9]

```
IF PEEK(49249) > 127 THEN PRINT "PB0 LINE IS HIGH"

IF PEEK(49250) < 128 THEN PRINT "PB1 LINE IS LOW"

IF PEEK(49251) <= 127 THEN PRINT "PB2 IS LOW"
```

The BASIC program PB.TESTER on the *Chaos in the Laboratory Programs #1* disk displays the status of the three switch input lines.

The PB0 and PB1 switch input lines on all Apple II computers except the II+ are also connected to the <OPEN APPLE> and <CLOSED APPLE> keys. The exact wiring of the keys is different on different computers, but the net effect is always that pressing the <OPEN APPLE> key causes the computer to act as if a high voltage was applied to the PB0 line. Pressing the <CLOSED APPLE> key causes the computer to act as if a high voltage was applied to the PB1 line. Connecting the "Apple" keys to the PB lines was a change made when the Apple IIe was introduced. Several other changes to the PB lines have occurred in the newer models of Apple II computers. The result is that the PB input lines are electrically different on all of the different models. Several of the projects in this book were carefully designed to work around these differences. In other cases, we suggest slightly different versions of the hardware for certain computer models (especially the Apple IIc and IIc+).

Using the PB0 and PB1 lines on computers other than the II+ can lead to some unexpected results. The most frequent problem involves the Apple's *automatic self-test*. If the PB1 line is connected to a voltage higher than two volts, the computer responds in the same way as it would if the <CLOSED APPLE> key was held down. The computer has a built in self-test routine that is started automatically each time the <CLOSED APPLE> key is held down when the computer is first turned on (or rebooted). During the self-test, the video monitor goes dark and the computer goes through several seconds of diagnostic tests. When the tests end, a status message is reported. Turning on (or rebooting) the computer while a voltage signal is being fed into the PB1 lead can accidentally initiate

---

[7]The numbers in the parentheses are the memory addresses in hexadecimal or base-16 notation. This number system is often used with computers. Hexadecimal numbers in this book are written preceded by a dollar sign to distinguish them from decimal numbers.

[8]In electronics terminology, switch input lines are referred to as TTL inputs.

[9]The status of the switch input lines can also be checked using machine language. Several examples are included on the *Chaos in the Laboratory Programs* disks.

this self-test. This can be frustrating and confusing to the user. We avoid the self-test problems in these projects by not using the PB1 input line very often.

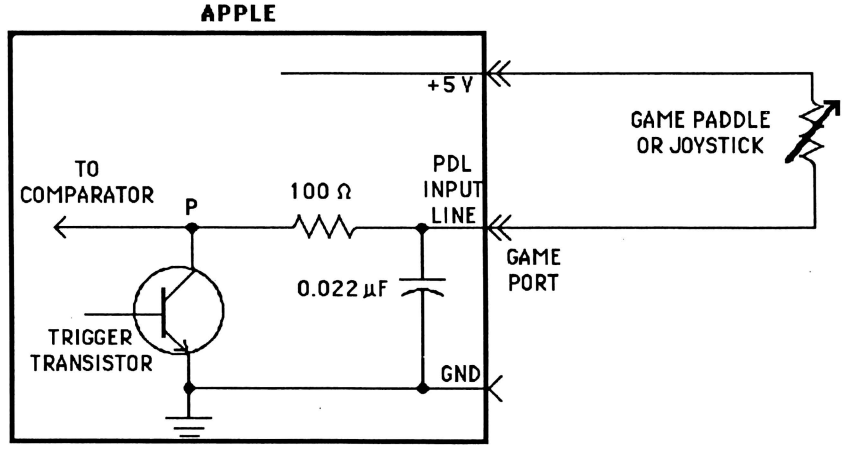
Paddle Inputs (PDL0, PDL1, PDL2, PDL3)

The PDL input lines were originally designed to be used to read the position of a game paddle (hence the label PDL) or joystick. There are four of these PDL lines on the Apple II+, IIe and IIGS game ports. They are labeled PDL0, PDL1, PDL2 and PDL3. The Apple IIc, IIc+ and Laser 128 computers have only the PDL0 and PDL1 lines.

The game paddle or joystick that is used with the PDL inputs is really a variable resistor. Its resistance varies from 0 to 150 kilohms. It is connected between the +5V line and one of the PDL lines on the game port. Using a command built into Applesoft BASIC, the resistance can be determined by the computer. The four BASIC commands PDL(0), PDL(1), PDL(2), and PDL(3) produce an integer between 0 and 255 that depends on the resistance of the variable resistor. The number in the parentheses determines the PDL line being checked. The larger the number displayed on the monitor, the higher the resistance of the variable resistor. Typical BASIC statements using these commands are shown below:

```
100 PRINT PDL(0) : REM DISPLAY THE STATUS OF PDL0
200 X = PDL(1) : REM SET X ACCORDING TO STATUS OF PDL1
```

To understand what is really going on when you use a BASIC PDL(0) command, we need to take a more detailed look at the PDL input lines. A typical circuit using the PDL lines is shown below:



Typical Game-Paddle Circuit

The part of the circuit drawn inside the rectangle is inside the computer.[10] An indirect method is used to determine the resistance connected to the PDL line. To understand this method, you need to understand how the various components of the circuit function:

- **variable resistor**: The variable resistor typically varies in resistance from 0 to 150 kilohms. It limits the flow of current from the +5V terminal toward the capacitor according to Ohm's law. If the resistance is increased, the current to the capacitor is reduced. If the resistance is decreased, the current increases.
- **0.022-µF capacitor**: A capacitor is a device that stores electrical charge. As the charge stored on the capacitor increases, the voltage across the two leads of the capacitor (and at point "P" in the diagram above) increases.
- **trigger transistor**: The trigger transistor in this circuit acts as a switch that is controlled by the computer. When the trigger transistor is "on," current can pass through it. When it is "off," no electrical charges can move through it.

[10]The circuit above is a PDL circuit for the Apple IIe. Other Apple II models have similar, but not identical, PDL input circuits.

- **comparator**: The comparator is an electronic circuit that can sense the voltage level at point "P" in the circuit diagram above. The computer uses the comparator to determine if the voltage at point "P" is less than or greater than a specific voltage level (3.3 volts).

Under normal circumstances, the trigger transistor is in its "on" condition. This allows any charges that might be present on the positive side of the capacitor to flow through the 100-Ω resistor to ground. As a result, the capacitor is normally totally discharged. The voltage at point "P" is 0 volts. When the BASIC PDL( ) command is executed, the trigger transistor is switched "off." At the same instant, the computer begins keeping track of time and checking the voltage at point "P". The current from the +5V line through the variable resistor now acts to charge the capacitor. The voltage across the capacitor (and at point "P") begins to increase. The computer measures the amount of time it takes for the voltage at point "P" to reach a specific level (as determined by the comparator). A machine-language routine in the Apple's ROM[11] is used to do this. The time is really just a "count" of how many loops the program goes through before the capacitor is charged. If the variable resistor is set for low resistance, then the current is large and the capacitor will charge up quickly, which means a low count is registered. If the variable resistor is set for a high resistance, the current into the capacitor is small and the capacitor will take a long time to become charged, which means a high count will be indicated. If the capacitor is not charged even when the count reaches 255, the routine ends and a count of 255 is registered. Each loop takes 11 µsec (11×10^{-6} sec). The count is always a whole number in the range 0 to 255. For example, a count of 193 means that it took 193 x 11 µsec (or about 2 milliseconds) for the capacitor to become charged. A count of zero indicates that the capacitor charged almost immediately, so the current must have been very large.

As you can see, the PDL circuit and accompanying BASIC commands are an elaborate way for the computer to sense the resistance of an external resistor. The BASIC program PDL.TESTER on the *Chaos in the Laboratory Programs #1* disk displays the status of the four PDL input lines.

Annunciator Outputs (AN0, AN1, AN2, AN3)

The Apple II+, IIe, and IIGS internal 16-pin game ports have four *annunciator output* connections. These four terminals are labeled AN0, AN1, AN2, and AN3. With simple BASIC statements, you can change the voltage at any of these four terminals from low (near 0 volts) to high (about 5 volts).[12] These terminals are **not** available on the Apple IIe 9-pin external game port.

The statements used to control the voltage at the annunciator terminals are shown in the table below:

Condition	BASIC POKE Statement
AN0 off (0 volts)	POKE 49240, 0
AN0 on (+5 volts)	POKE 49241, 0
AN1 off	POKE 49242, 0
AN1 on	POKE 49243, 0
AN2 off	POKE 49244, 0
AN2 on	POKE 49245, 0
AN3 off	POKE 49246, 0
AN3 on	POKE 49247, 0

Notice that the first number in each of these POKE statements is different for each line. The second number in the POKE statement is just a filler. A zero is used in all of the statements above, but any

[11] *ROM* is an abbreviation for "read only memory." It consists of memory chips in which computer code is permanently stored. This code can be read by the computer at any time, but it can never be rewritten.

[12] The annunciator lines are digital outputs. This means that they can have one of two possible values. Project 10 in this book is a circuit to control an analog voltage from the game port. Analog outputs can vary over a range of many possible voltages.

integer between 0 and 255 could be used. It makes no difference, but there must be some number there.[13] The sample BASIC program below would turn the AN0 terminal on and off at regular intervals:

```
10 POKE 49241, 0 : REM AN0 ON
20 FOR J= 1 TO 2000 : NEXT J : REM STALL
30 POKE 49240,0 : REM AN0 OFF
40 FOR J= 1 TO 2000 : NEXT J : REM STALL
50 GOTO 10 : REM REPEAT
```

Be careful when using the annunciator lines. The Apple's annunciator lines cannot be used to power anything that requires very much current. If you try to draw too much current from an AN line or sink too much current into it, you could actually damage the computer.[14] The annunciator lines are designed to drive only the inputs of other integrated circuits.[15] If a larger load is to be controlled with these lines, a secondary "buffer" circuit must be used between the annunciator line and the load. The annunciator output lines are used in this way in several of the projects in this book.

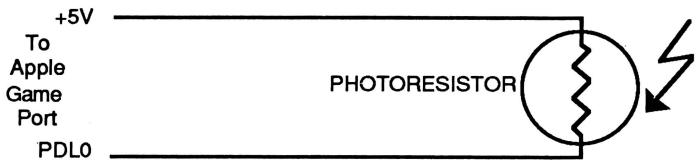
The AN3 line is used on an Apple IIGS to control whether the double high-resolution graphics are used. This can cause some complications when using some of the projects in this book. This topic is discussed in chapters that use the AN3 line. On some old video display cards, other annunciator lines were used for controlling the video. Programs changing the status of the annunciator lines may cause unusual graphics effects.

Strobe Output

The 16-pin internal game port on an Apple II+ or IIe has an infrequently used strobe-output line. This connection is not available on the 9-pin external game port. It is also not available on the IIGS 16-pin game port. The strobe-output line is normally at 5 volts. The voltage on the line drops briefly (for about 1 µsec) each time the memory location 49216 is used (either to read or to store a number). None of the projects in this book use this connection.

A Simple Game Port Interfacing Example

Just to show you how extremely simple a laboratory interfacing project can be, consider the following circuit:



Simple Light Sensor Circuit

This circuit consists of a single component, a *photoresistor*, connected to the game port. Photoresistors are light-sensitive resistors. These are small, light-dependent resistors usually made of cadmium sulfide (CdS) or cadmium selenide (CdSe). Photoresistors are inexpensive and are sold at Radio Shack stores (#276-1657 or #276-118). As the amount of light shining on the photoresistor changes, the resistance changes and the computer's PDL(0) command will read a different value. A

[13]This is a fairly common, but strange way of doing things on the Apple II. It is called a *soft switch.*

[14]As is the standard practice in electronics, by *current* we mean the movement of **positive** charge. When the annunciator output line is high (5V), it acts as a *source* of current. When the line is low (0 V), it acts as a *sink* for current.

[15]The annunciator output lines on an Apple II+ are standard 74LS TTL outputs. The annunciator lines on an Apple IIe or IIGS connect directly to custom ICs. The Apple reference manuals warn that these lines are to be used to drive only one TTL load.

circuit of this type is used in two of the projects in this book: Project 3 - Computer Vision and Project 4 - Colorimeter.

If possible, connect a photoresistor to the Apple II game port and try out this simple circuit. You can use any model of Apple II computer, and you can use either a 9-pin or a 16-pin game port.

It turns out that the software needed to do this lab interfacing project is about as simple as the hardware. A complete working program to read the status of the photoresistor is listed below:

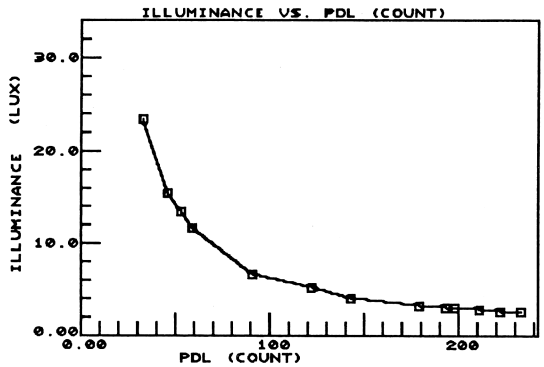
```
10 PRINT PDL(0)
20 GOTO 10
```

Type in this program and then type RUN and press <RETURN>. You should get a display of counts that change as the light striking the photoresistor changes. To stop the program, press the <CONTROL> key and then the <C> key.

The combination of the circuit above and this program will allow you to use your Apple II as a light meter. The spectral response of this meter nicely matches the human eye. It is extremely sensitive. The count displayed represents the resistance of the photoresistor and will decrease as the light intensity increases. This system can be used for many experiments where it is only important to show whether the light intensity is increasing or decreasing. Examples include:

- comparing light levels in different regions
- studying the spacing of bright regions in a diffraction or interference pattern
- determining when the sun sets as part of a weather station
- studying the period of a variable star
- measuring changes in light levels as cloud conditions change
- measuring the light transmitted through a solution
- measuring the light reflected off various objects

This circuit and the simple software have some limitations. The count displayed does not relate in a simple way to the illumination as it would be measured by a standard light meter. The graph below shows the response of a typical photoresistor vs. illuminance measurements made with a calibrated light meter.[16]



It would be nice if the program displayed illuminance instead of just a count that represents resistance. If you make a thorough study of how photoresistors respond to light, you can probably develop a mathematical equation that will take a count reading representing resistance and convert it to an illuminance reading. You do this by analyzing graphs like the one above or using a computer to do a mathematical analysis of a number of experimental measurements. For the Radio Shack #276-118 photoresistor, here is an equation we have developed that works fairly well:

```
I = 1 / PDL(0) *780
```

[16] *Illuminance* is a measure of the relative intensity of light, adjusted to match the response of the human eye.

If you add two lines to the simple program you just typed in, it will display illuminance (in units of lux) as well as the resistance (in counts). If you still have the program in the computer's memory, all you have to do is type lines 15 and 16.

```
10 PRINT PDL(0)
15 I = 1 / PDL(0) *780
16 PRINT I; " LUX"
20 GOTO 10
```

When you RUN this program it should display both the count and the illuminance in lux. The illuminance readings will be approximate. For the best accuracy, we should adjust the equation slightly (for example, change the value 780) to best match a particular photoresistor. This procedure is called *calibration*. Most programs that involve measurement have some provision for calibration by comparing the computerized sensor with a traditional sensor. A good example is the thermistor program in this book. A thermistor is a temperature-sensitive resistor. In that project you prepare three water containers at three different temperatures and use them for calibration. In each case you measure the water temperature with a thermometer as the computer takes a reading. After all three readings are made, the computer program automatically adjusts the constants of the calibration equation to best match the thermistor being used.

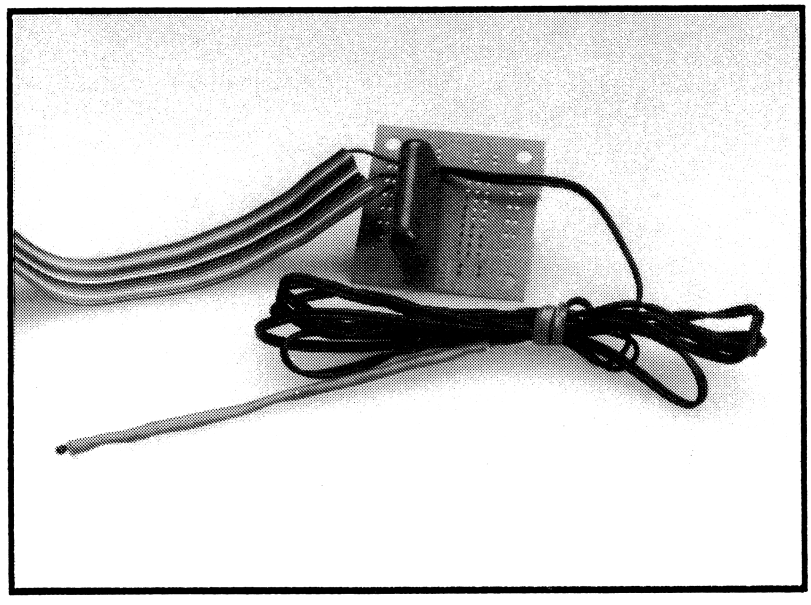
Several of the projects in this book (Barometer, Magnetic Field Sensor, and Light Meter) use electronic circuits that were designed especially to produce a linear calibration relationship. This means that if you graphed the voltage produced by the sensor vs. applied signal values, you would get a straight-line graph. When the sensor produces a linear response, calibration of the probe is easier.

Since most people do not have a light meter that reads in illuminance units, for your light meter you may take the equations as listed above and accept that your illumination values are approximate. Note that in many cases we are interested in relative illumination readings, and the absolute illumination in lux is not as important.

The other limitations of this simple light meter are:

- It is useful only over a narrow range of light levels (it works only at low light levels).
- Photoresistors become less sensitive with time, especially if they are exposed to ultraviolet light, high temperatures, or high humidity.
- Photoresistors often change their response, depending on the light level they were exposed to previously. For example, a photoresistor that was stored in the dark will be slightly more sensitive than an identical photoresistor that has been exposed to bright light for several minutes.
- The sensitivity of photoresistors change noticeably with temperature.
- Photoresistors have a slow response time; that is, it takes them a tenth of a second or so to respond. This is not important for most experiments, but in some cases (for example, if you are trying to show the flicker of a 60-Hz AC light source), it may cause problems.

Because of these limitations of the circuit discussed above, we have developed a somewhat more complicated Light Meter as Project 6 of this book.



Project 1

Thermistor Temperature Probe

This project consists of building a computerized temperature measurement system. The temperature probe is quick to respond and is very accurate over the temperature range -30°C to 70°C. It can be used up to 100°C. The system uses a *thermistor,* a resistor that changes resistance as temperature changes. Programs on the *Chaos in the Laboratory Programs #1* disk allow the computer to measure the resistance of the thermistor and convert it to a temperature.

Project Requirements

Computer

This project can be done with any model of Apple II computer.

Background Knowledge

- Electronics: This is the easiest project in this book to build. To fully understand the operation of the thermistor temperature probe system, you need to be familiar with what capacitors and resistors are and what they do. You should also understand Ohm's law and the terms *charge*, *current*, and *voltage*.
- Programming: None required for the core project. Several of the project extensions involve adding features to the BASIC program.

Parts

- ☐ A negative temperature coefficient (NTC) thermistor with a resistance at room temperature of about 10 kΩ. The thermistor should be physically small so that it will respond quickly to temperature changes. The following thermistors are recommended. Note that the glass-enclosed thermistors can be used in harsh chemicals that might destroy an uncoated thermistor.

Vendor	Part Number
Radio Shack	271-110 ($2)
Arrow Electronics	2248AAD (Fenwal FA41J1WC) ($2)
Arrow Electronics	2248AAA (Fenwal GB41P2) ($4)
Newark Electronics	30F1725 (Fenwal GB41J1) ($8, glass-coated bead)
Newark Electronics	30F1770 (Fenwal FA41P2) ($6, glass probe)
Digi-Key	PNT119 (Panasonic, $1, large)
Mouser Electronics	ME334-4127-103 ($1, large)

- ☐ 4.7-μF capacitor, 10 volts or more (You can use almost any type of 4.7-μF capacitor, but some types of capacitors are much better than others in terms of temperature stability. The more stable this capacitor is, the more accurate your system will be. Here is a list of possible types of capacitors in order of preference, the best listed first: polycarbonate, polypropylene, polyester, Mylar, tantalum, electrolytic).
- ☐ circuit board (Almost any small circuit board will work. One possibility is one-half of a Radio Shack #276-159.)
- ☐ cable to Apple game port (see *Appendix A*)
- ☐ 3/16 inch heat-shrink tubing, approximately 10 cm long
- ☐ 2-conductor cable, 1 or 2 meters long (Any flexible, multi-stranded wire can be used.)
- ☐ 1/16 inch heat-shrink tubing, 3 cm long, for insulating one connection

Note: A "Thermistor Parts Kit" containing all of the parts listed above (with a metalized polyester capacitor) is available from Vernier Software for $10.00. Two versions of this kit are available. Order Code CTP-16 includes a cable to connect to the 16-pin internal game port. Order Code CTP-9 includes a cable to connect to the 9-pin external game port. The thermistor used in this kit is not enclosed in glass. This kit does not include the sealant needed to waterproof the probe. An order form is in the back of this book.

Tools and Supplies

Low-power soldering iron or soldering pencil, solder, wire cutters, wire strippers, thermometer, and silicone sealant (GE Silicone II or equivalent, available at hardware and discount stores). A hair dryer is very helpful when you use the heat-shrink tubing.

Background Information

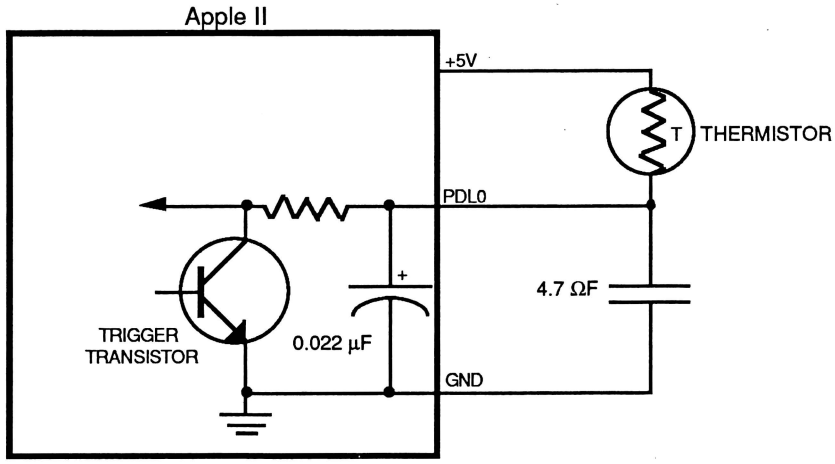
The word *thermistor* is short for "temperature sensing resistor." Thermistors come in many shapes and sizes. They are used for several purposes in electronics circuits, including temperature control, producing time delays, and surge suppression. We will be using what is called a bead thermistor.

Bead Thermistor (magnified about 10 times)

Bead thermistors are specifically designed for use as temperature measurement devices. Their small size means they can change temperatures quickly. These thermistors have a high negative temperature coefficient; that is, their resistance decreases quite a lot for a relatively small increase in temperature. We will use the computer to measure the resistance of the thermistor and then calculate the temperature that corresponds to this resistance.

To measure the resistance of the thermistor, we will use the Apple II game port PDL lines. The PDL lines are normally used for game paddles and joysticks. Their operation is explained in detail in "The Apple Game Port" section of the *Introduction*. We recommend that you take a few moments to read that section if you are not familiar with how the computer normally senses the condition of the PDL lines.

The way we will connect the thermistor to the Apple is shown below. The components drawn inside the rectangle are inside the computer.

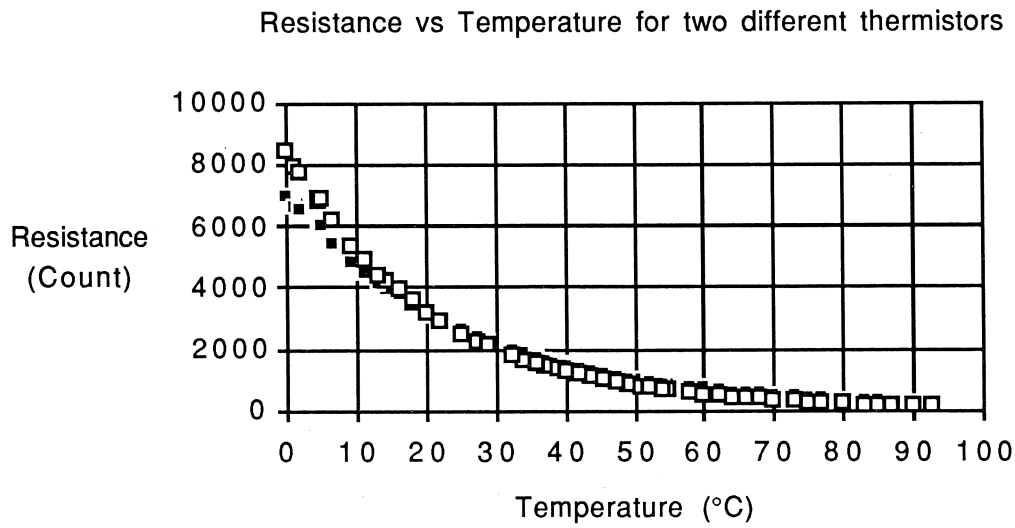


Thermistor Temperature Probe System

With this system, the thermistor takes the place of the variable resistor that is normally connected to the PDL0 line. The current through the thermistor is used to charge the computer's internal capacitor and an external 4.7-µF capacitor. The status of the PDL0 line is detected by measuring how long it takes the capacitors to charge up to 3.3 volts after the trigger transistor is turned off. If the resistance of the thermistor is high, it will take a long time to charge the capacitors. If the resistance is low, the charging will occur quickly. In fact, the time taken to charge the capacitors is proportional to the resistance of the thermistor.

This kind of timing is normally done with the Applesoft BASIC command PDL(0). The computer counts how many cycles of a timing loop occur before the voltage across the 0.022-µF capacitor reaches 3.3 volts. When the Applesoft BASIC command PDL(0) is used to measure the charging time, an integer "count" ranging from 0 to 255 is reported. This count is a measure of resistance. This method of timing does not provide enough precision for this project, so we will use a slightly more elaborate timing system. We will use a special machine-language program called PDL.CHECK. This routine reads the time it takes the capacitor to charge in a similar way, but it can count for a much longer time. The PDL.CHECK program counts from 0 to 40,960, with each count representing 18 microseconds. It can measure charging times of up to 0.7 seconds, which is plenty of time for the current through the thermistor to charge the capacitors. With this new timing routine and the thermistor we will be using, the count will range from about 7000 to about 100, as the resistance of the thermistor changes. We now can accurately measure the resistance of the thermistor as a count reported by the computer. "Counts" can be considered an alternative unit of resistance (instead of the normal ohms).

Unfortunately, the relationship between the thermistor's resistance and its temperature is not a simple one. The graph below shows the response of two different bead thermistors of the type that we recommend for this project.



Experimenters and mathematicians have tried to find a mathematical equation that exactly fits this type of graph. Several different equations have been used.[1] We will use the Steinhart-Hart equation. This equation closely matches the response of thermistors of the type recommended for this experiment.[2]

$$1/T = A + \ln(R) * B + (\ln(R))^3 * C \quad \text{Steinhart-Hart equation}$$

where: ln (R) is the natural (base e) log of the resistance (measured as a count from the computer)
T is the temperature in kelvins
A, B, and C are constants for the particular thermistor

To calibrate a particular thermistor, you need to determine the values of the constants A, B, and C for that thermistor. To do this we will measure the resistance (in counts) of the thermistor at three known temperatures. This information can then be used to solve for the three constants. This involves a little bit of mathematics, but fortunately, the THERMISTOR program on the *Chaos in the Laboratory Programs #1* disk can do the calculation quickly. The mathematics involved are explained in the "Supplemental Information" section at the end of this project. Other empirical equations that are used with thermistors are also discussed.

The program THERMISTOR on the *Chaos in the Laboratory Programs #1* disk includes an option for calibrating the thermistor and another mode for displaying the temperature reading. Once the thermistor probe is calibrated, its calibration data are saved on disk and can be used the next time you start up the program.

One of the suggested project extensions is the construction of a second thermistor probe. Many of the other suggested project extensions involve adding new features to the THERMISTOR program.

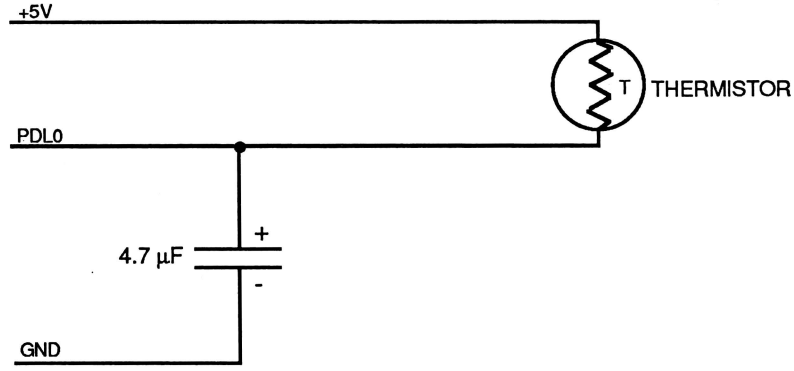
---

[1]Equations like this are called *empirical equations*. Empirical equations are developed to coincide with experimental data, as opposed to theoretical equations, which are developed to coincide to a scientific theory.

[2]For more information on this equation, refer to "An Empirical Mathematical Expression for Resistance-Temperature Relationship of NTC Thermistors" by Henry A. Hobbs, Fenwal Electronics, 1982.

Step-by-Step Instructions

The complete schematic for the Thermistor Temperature Probe is shown below:

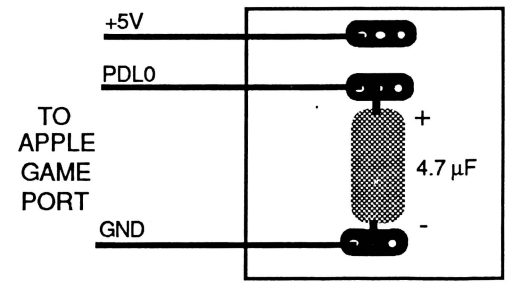


Schematic of Thermistor Temperature Probe

☐ 1. **Solder the capacitor to the circuit board.** Insert the leads of the 4.7-µF capacitor through the circuit board. The capacitor should be on the side with no copper. In this case it does not matter too much which holes you use. If the capacitor you are using is polarized (one side is marked with a "+" or "–" sign), make note of which are the "+" and "–" leads. The capacitor included in the Vernier Software "Thermistor Temperature Probe Parts Kit" is not polarized. Solder the capacitor to the plated side of the circuit board. The leads of the capacitor are thick and may take a while to heat up—be patient in your soldering. Cut off excess leads after soldering.

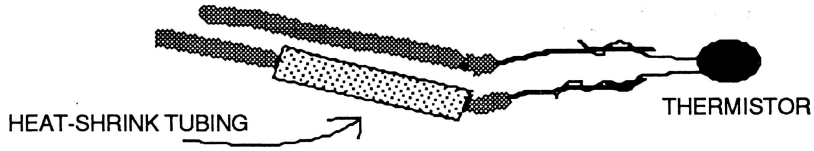
☐ 2. **Prepare the cable to the Apple game port.** Refer to *Appendix A* for information on cables. Either a 9-pin or a 16-pin cable can be used. Only the following leads will be connected: +5V, PDL0 and GND. Strip a few millimeters of insulation from the ends of these three leads.

☐ 3. **Connect the game port cable leads to the circuit board.** Solder the PDL0 line of the cable to the "+" lead of the capacitor (or either lead, if your capacitor is not polarized). Solder the GND line to the other lead of the capacitor. Solder the +5V line to an isolated spot on the circuit board. Later, one of the leads from the temperature probe will connect to this spot.

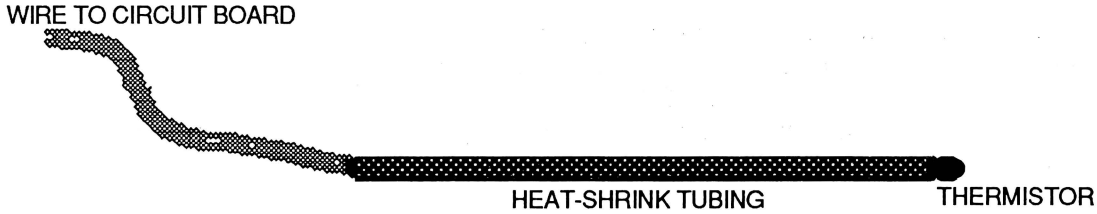


☐ 4. **Solder the 2-conductor cable to the thermistor.** Strip a centimeter of insulation from each of the ends of the wires on one end. Use small-diameter (1/16 inch) heat-shrink tubing for insulating one of the thermistor leads. Separate the wires for a few centimeters and place the tubing over one of the wire leads before you start to solder. Twist one of the thermistor leads around the wire lead with the heat-shrink tubing. Solder this connection. Move the heat-shrink tubing down until it just touches the body of the thermistor. It should completely cover the exposed metal of one of the thermistor leads. You can shrink the tubing by heating it in either of two ways: use a hair dryer, or rub the soldering iron gently against it. Now solder the other lead of the thermistor to the other wire lead. Note that heat-shrink tubing is not required for this second connection.[3]

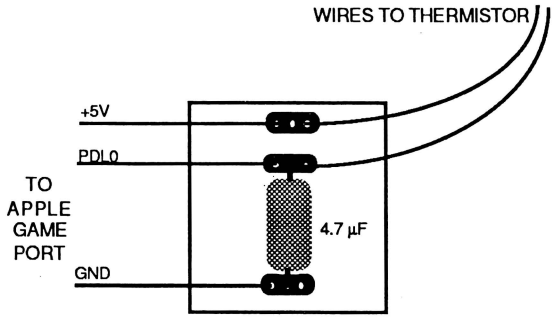
[3]In fact, if you use heat-shrink tubing on both leads, you may not be able to fit the thermistor leads into the body of the probe tube in the next step.



☐ 5. **Complete the temperature probe.** Slip the 2-conductor cable, with the thermistor attached, through the 3/16 inch heat-shrink tubing. Slide the tubing along the wire until it touches the thermistor. Put a small amount of silicone sealant inside the tube to guarantee a watertight seal. Remove the excess silicon sealant. Now warm the heat shrink tubing with a hair dryer to tighten it around the wire and the thermistor. It is important to get a watertight seal around the leads of the thermistor. Remember that the choice of sealant used here affects the chemicals in which the temperature probe can be used. We have found Silicone II from General Electric to be very durable, but there are some chemicals that will dissolve it; notable examples are cyclohexane, naphthalene, and paradichlorobenzene. Place sealant at the other end of the heat shrink tubing as well. This prevents liquids from getting down the tube if the probe is totally submerged.



☐ 6. **Connect a thermistor probe to the circuit board.** Strip the insulation from the leads of the end of the two-conductor cable and solder them to the circuit board. One wire should be connected to the +5V lead from the computer. The other wire connects to the PDL0 lead from the computer (which is also connected to the capacitor). It does not matter which one connects to which spot. Refer to the diagram below:



☐ 7. **Test the Thermistor Temperature Probe**. Turn off the computer and connect the circuit to the game port. Place the *Chaos in the Laboratory Programs #1* disk in the disk drive and turn on the computer. If the computer acts strangely or will not turn on, flip the power switch off and double-check your circuit—you must have an error in wiring, probably involving the +5V line.

If the disk starts up normally, run the THERMISTOR program. When the menu is presented, choose the C - DISPLAY COUNTS mode. The computer should begin to display the count reading. It should be between 3500 and 2000. Touch and warm the thermistor and see if the count changes. The count should get lower as the temperature increases. If these tests are normal, you have a working probe system and you are ready for calibration. If the probe is not working properly, refer to the "Troubleshooting" section below.

☐ 8. **Calibrate the probe.** Select Z - CALIBRATION from the menu of the THERMISTOR program. For the first calibration, we recommend you select three temperatures that cover the entire range of the thermistor temperature probe system. The system can be used to measure temperatures from -30°C to 100°C. Calibrate at roughly 10°C, 40°C, and 70°C. Later, you can improve accuracy by calibrating over a narrower temperature range.

Be sure the silicon sealant has set before placing the probe in water.

Place the probe in a container of water at the first calibration temperature. Stir the water and measure the temperature of the water in °C with a good thermometer. Leave the probe and the thermometer in the water bath long enough for them to reach a steady reading. Press the <RETURN> key and then enter the actual temperature in °C. Repeat this process in water at two other temperatures. The program calculates the value of the three calibration constants and saves the values on the disk in a file called THERMISTOR.CAL. From this time on, whenever the THERMISTOR program is used these calibration values will be loaded into memory so that the program can calculate actual temperatures from the count readings. If you ever want to recalibrate, simply select the Z - CALIBRATION mode again.

☐ 9. **Check the accuracy of the completed Thermistor Temperature Probe.** Select T - DISPLAY TEMPERATURE from the menu of the THERMISTOR program. The program should continuously display the count and the temperature reading in degrees Celsius.[4] Check the accuracy of the temperature probe at ten different temperatures over the range 0°C to 100°C. Compare the readings from the probe with thermometer readings. Is the temperature probe accurate over the entire temperature range? Determine the maximum error; that is, the largest difference in readings between the probe and your thermometer. Check to see if the probe readings are reproducible; that is, if the probe is moved out of and back into the same temperature several times, does it return to the same value? Refer to the "Supplemental Information" section below for tips on using the thermistor probe.

Troubleshooting

Check the following if your temperature probe system is not working properly:

◊ If the computer will not start up, there must be a short circuit (direct connection between the +5V and the GND leads).
◊ If the count displayed by the DISPLAY COUNTS mode is -24576, the probe leads are not connected properly. Check for a loose connection on the circuit board or a loose wire inside the probe.
◊ If the count displayed by the DISPLAY COUNTS mode is 1, the probe is shorted. Check for a connection between the thermistor leads inside the probe.
◊ If a probe reading changes inconsistently, check for water inside the thermistor probe.

If none of this helps:

◊ Make sure you are using the leads from the Apple game port properly. Use an ohmmeter to check the wiring from the game port pins to the connections on the circuit board. Are you using the PDL0 line and not one of the other PDL lines?
◊ If you are using a 16-pin game port cable, is it inserted with pin #1 in the right spot? Are all 16 pins inserted in the Apple II game port socket? Is there a bent or broken pin?
◊ If you used a polarized capacitor, make sure the capacitor is oriented properly. The "+" lead should be connected to PDL0. Polarized capacitors are not used in the Vernier Software Thermistor parts kit.

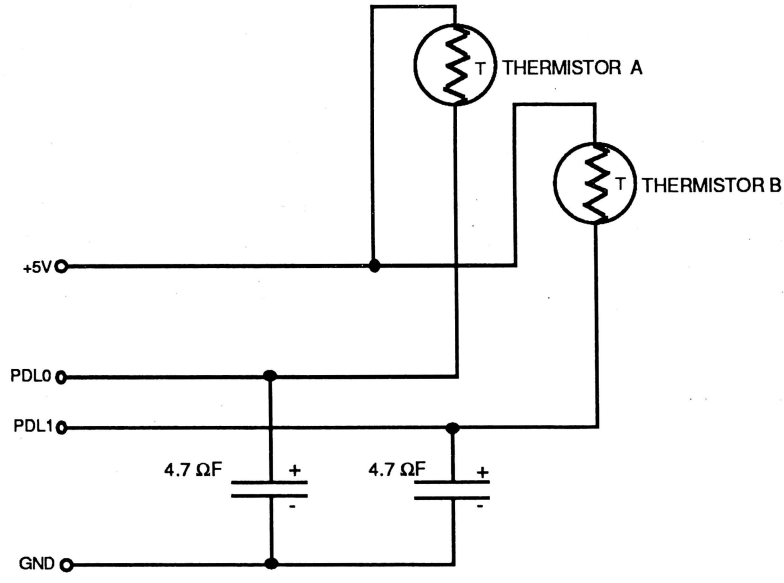
---

[4]Notice that too many digits are used when the temperature is displayed. The temperature should probably be displayed to the nearest 0.1 degree or 0.01 degree. Improving this part of the program is left as one of the project extensions.

Project Extensions

Circuit Modifications

◊ Add a second thermistor temperature probe to the system. An additional temperature probe can be added to the circuit using the PDL1 line. The schematic is shown below:



A 2-Thermistor System

This circuit is essentially a double version of the circuit described earlier. To use both probes, you will have to use the THERMISTOR.2 program that is included on the *Chaos in the Laboratory Programs #1* disk. This program is similar to the THERMISTOR program, except that it reads both probes.

◊ Modify the thermistor probe so that it can be used in harsh chemicals. Probes built following the step-by-step instructions are waterproof and safe to use in some chemicals. However, there are some chemicals that dissolve silicone sealant or the heat-shrink tubing. You may want to build a more chemical-resistant thermistor probe. One way to do this is to mount the thermistor inside a small glass test tube or a glass tube with the end fused closed.

Note that there is one disadvantage to this kind of probe construction: The glass will insulate the thermistor from the outside world and make it slower to respond to temperature changes. Try to think of other ways to make a tough, chemical-resistant probe.

◊ Some thermistors can be used at very high temperatures. Contact thermistor suppliers and obtain one of the high-temperature thermistors. Can these thermistors be calibrated in the same way as the ones used in the core project?

Program Improvements

There are a number of improvements you might want to make to the THERMISTOR program. The "Program Design Notes" section below explains how the THERMISTOR program operates. It should help you in making modifications. Some suggested changes are given below:

◊ Round the temperatures displayed to the nearest 0.01°C or 0.1°C.
◊ Change the program so it can collect data in memory and display it on the video monitor or print it later.
◊ Change the program so that it displays the temperatures in degrees Fahrenheit, degrees Celsius, or kelvins.
◊ Add "real-time graphing" to the program; that is, have it make a graph as the temperature data is taken.
◊ Allow the temperature data to be saved on a disk.
◊ Have the program display the time of each temperature reading measured from the start of the readings. Notice that each temperature measurement takes about one second.
◊ If your computer has a "clock card" to keep track of real time, change the program so that it reads the clock time and stores it along with the temperature reading. This would be very useful for a weather station.
◊ Change the program so that temperature readings at preselected time intervals (for example, every hour) are saved in memory. Again, this would be desirable for a weather station.
◊ Program the computer to record the maximum and minimum temperature reading.
◊ Change the program so that the computer makes a warning noise whenever the temperature is outside a specified range. Expand this idea so that the computer uses optically-isolated switches to turn on heaters, lights, or motors when the temperature reaches certain limits. This project will require a lot of experience with electronics circuits, as well as programming.
◊ Add an LED connected to one of the annunciator outputs of the game port so that it lights each time the temperature is outside a specified range.
◊ Add a mode to the program that graphs the temperature readings versus time.
◊ Add a mode to the program that displays the temperatures in large, block digits so that they can be read at a distance. Use the QUIZ program on the same disk as a model. This program uses a machine-language routine to draw large characters on the monitor.
◊ If you have modified the circuit to allow the use of two thermistors, change the THERMISTOR.2 program so that it can be used as a humidity meter. Cover one probe with a wet cloth and leave the other one dry. Have the computer read the two temperatures, calculate the difference, and then determine the relative humidity.

Additional Experiments

The Thermistor Temperature Probe can be used for a number of experiments:

◊ Plotting cooling curves
◊ Measuring specific heats and heat capacity
◊ Testing for changes in the freezing point of water when various chemicals are added
◊ Studying temperature at various heights in a room
◊ Measuring rates of cooling and the efficiency of insulation
◊ Measuring heat of reaction
◊ Studying outdoor temperature variation
◊ Studying indoor temperature, thermostat function, furnace cycles, etc.
◊ Studying thermal gradients in a container of water, a lake, or river
◊ Measuring the latent heat of fusion of ice

Supplemental Information

Suggestions for Using the Thermistor Temperature Probes

Calibration – Careful calibration of the probes is important for accurate temperature readings. Always use a good thermometer. The accuracy of the probes can be only as good as the device with which you calibrate. Make sure that the thermometer and the thermistor are really at the same temperature. For example, if you are calibrating using water, make sure that both the probe and the thermometer are at the same level in the container. There can be a difference of several degrees at

different positions in a container of water. Stirring the water vigorously helps keep the temperature uniform.

Choice of calibration temperatures – The resistance of the thermistor closely matches the thermistor equation. Of course, there is some deviation from a perfect fit. The initial calibration suggested in the "Step-by-Step Instructions" was at 10°C, 40°C and 70°C. This calibration should yield fairly good accuracy over the whole range of the thermistor. The probe will read most accurately if it is calibrated at temperatures similar to the temperatures to be encountered. For example, if you are going to use the probe to measure indoor air temperatures, it would be best to calibrate it at 15°C, 20°C and 25°C. If you are going to study human mouth temperatures, it would be best to calibrate at 35°C, 37°C and 39°C.

Using probes in chemicals – Be careful when using the temperature probes in chemicals. Some chemicals may react with the sealant, the heat-shrink tubing, or the thermistor itself. Cyclohexane and other nonpolar solvents will dissolve silicone sealant.

Wet probes – If you start getting strange results from a probe, a likely cause is liquid inside the probe tube. The liquid conducts electricity itself and acts like a second resistance in parallel with the thermistor. The probe resistance will be decreased and the temperature reported will be too high. If the probe leaks, dry it out and reseal it.

Calibration change due to changes in room temperature – If you are using 4.7-µF capacitors that have poor temperature stability (for example, electrolytic capacitors), you may need to recalibrate if the temperature of the room in which you are working changes significantly.

Solving the Thermistor Equation

The equation used in the THERMISTOR program is called the Steinhart-Hart equation. It is an example of an *empirical* equation. An empirical equation is developed by collecting experimental data (in this case, resistance vs. temperature for a thermistor) and trying to find the mathematical relationship that best matches the data. Many other equations have been suggested for use with thermistors. Several of them are listed later in this chapter. The Steinhart-Hart equation works well for many thermistors, and is still simple enough that calibration and use of the thermistor is not too difficult. In our tests, using thermistors calibrated at 11.2°C, 40.0°C and 70.0°C using the Steinhart-Hart equation, our thermistor temperatures never differed from temperatures measured with a thermometer by more than 1.0°C in the temperature range 0 to 90°C. The accuracy decreases at higher temperatures.

This section explains how the Steinhart-Hart equation can be used to calibrate a thermistor and then shows how to calculate the temperature from a resistance reading.

$$1/T = A + \ln(R) * B + (\ln R)^3 * C \qquad \text{Steinhart-Hart equation}$$

where: R is the resistance of the thermistor in ohms
T is the temperature in kelvins
A, B, and C are derived coefficients that depend on the particular thermistor

When you are calibrating a thermistor, you measure the resistance of the thermistor (R_1, R_2, and R_3) at three different known temperatures (T_1, T_2, and T_3). You assume that the Steinhart-Hart equation holds true at all three of these temperatures, therefore:

$$1/T_1 = A + \ln R_1 * B + (\ln R_1)^3 * C$$
$$1/T_2 = A + \ln R_2 * B + (\ln R_2)^3 * C$$
$$1/T_3 = A + \ln R_3 * B + (\ln R_3)^3 * C$$

If you treat A, B, and C as unknowns, you have three simultaneous equations with three unknowns. In more standard form, the equations become:

$$A + m_1 * B + n_1 * C = k_1$$
$$A + m_2 * B + n_2 * C = k_2$$
$$A + m_3 * B + n_3 * C = k_3$$

with $m_1 = \ln R_1$ and $n_1 = (\ln R_1)^3$ and $k_1 = 1/T_1$ and similar substitutions for the other two equations. These equations can be solved in a number of different ways. The example below uses determinants.

Using determinants to solve these simultaneous equations:

$$A = \frac{\begin{vmatrix} k_1 & m_1 & n_1 \\ k_2 & m_2 & n_2 \\ k_3 & m_3 & n_3 \end{vmatrix}}{\begin{vmatrix} 1 & m_1 & n_1 \\ 1 & m_2 & n_2 \\ 1 & m_3 & n_3 \end{vmatrix}}$$

$$B = \frac{\begin{vmatrix} 1 & k_1 & n_1 \\ 1 & k_2 & n_2 \\ 1 & k_3 & n_3 \end{vmatrix}}{\begin{vmatrix} 1 & m_1 & n_1 \\ 1 & m_2 & n_2 \\ 1 & m_3 & n_3 \end{vmatrix}}$$

$$C = \frac{\begin{vmatrix} 1 & m_1 & k_1 \\ 1 & m_2 & k_2 \\ 1 & m_3 & k_3 \end{vmatrix}}{\begin{vmatrix} 1 & m_1 & n_1 \\ 1 & m_2 & n_2 \\ 1 & m_3 & n_3 \end{vmatrix}}$$

Solving each of the determinants is a bit of a job. As an example, we will solve for the determinant that is the denominator in all three equations.

Subtracting row 1 from rows 2 and 3, column by column, and placing the new rows in rows two and three we have:

$$\begin{vmatrix} 1 & m_1 & n_1 \\ 0 & (m_2-m_1) & (n_2-n_1) \\ 0 & (m_3-m_1) & (n_3-n_1) \end{vmatrix}$$

$$= 1 * \begin{vmatrix} (m_2-m_1) & (n_2-n_1) \\ (m_3-m_1) & (n_3-n_1) \end{vmatrix}$$

$$= (m_2-m_1) * (n_3-n_1) - (m_3-m_1) * (n_2-n_1)$$

Similar procedures used on the other determinants lead to:

$$C = \frac{(m_2-m_1) * (k_3-k_1) - (m_3-m_1) * (k_2-k_1)}{(m_2-m_1) * (n_3-n_1) - (m_3-m_1) * (n_2-n_1)}$$

$$B = \frac{(k_2-k_1) * (n_3-n_1) - (k_3-k_1) * (n_2-n_1)}{(m_2-m_1) * (n_3-n_1) - (m_3-m_1) * (n_2-n_1)}$$

$$A = \frac{k_1 * m_2 * n_3 + m_1 * k_3 + n_1 * k_2 * m_3 - k_3 * m_2 * n_1 - k_2 * m_1 * n_3 - k_1 * m_3 * n_2}{(m_2-m_1) * (n_3-n_1) - (m_3-m_1) * (n_2-n_1)}$$

All of these calculations look complicated, but they take only a few seconds on the Apple II. Also, remember that these calculations need only be done when the thermistor is calibrated. These equations do make up a large part of the BASIC program THERMISTOR.

Fortunately, once this calibration is complete, the calculation of a temperature is much simpler. After calibration, the values of A, B and C are known, and the Steinhart-Hart equation is used to calculate the temperature from the resistance measured as follows:

$$1/T = A + \ln(R) * B + (\ln R)^3 * C$$
$$T = 1 / (A + \ln(R) * B + (\ln R)^3 * C)$$

This gives the temperature in kelvins. To convert to Celsius, 273.15° is added.

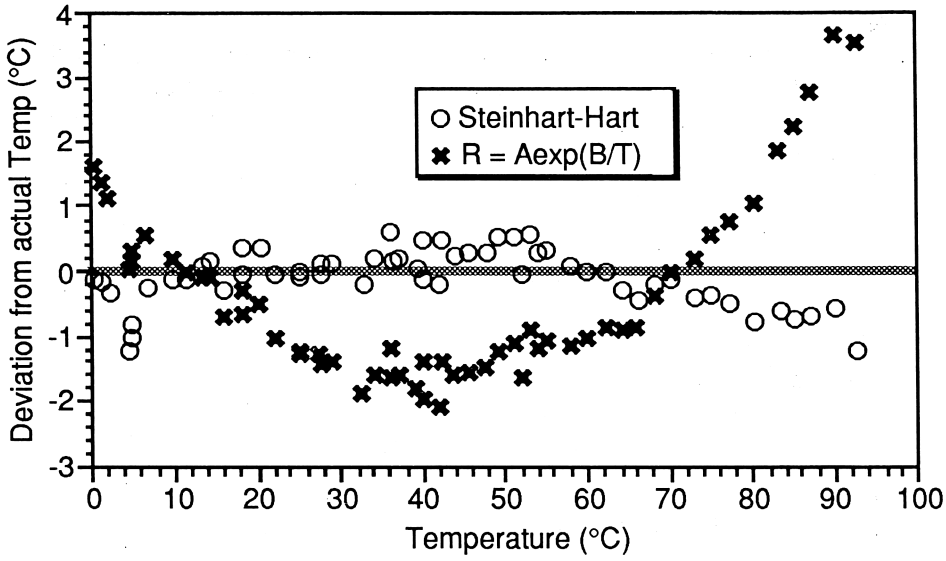
Other Thermistor Equations

Sometimes other empirical equations are used to convert thermistor resistance readings to temperatures. Some of them are simpler than the Steinhart-Hart equation used in this project. Several equations that are used for thermistor calibration are listed below. In all cases, R is resistance in ohms and T is the temperature in kelvins.

$1/T = A + \ln(R) * B + (\ln R)^3 * C$	Steinhart-Hart equation
$R = A \exp(B/T)$	equation 2
$R = R_0 \exp(B(1/T - 1/T_0))$	equation 3
$R = \exp(A_0 + A_1/T + A_3/T)$	equation 4
$R = \exp(A_0 + A_1/T + A_2 T^2 + A_3/T^3)$	equation 5
$1/t = a_0 + a_1 \ln R + a_2 (\ln R)^2 + a_3 (\ln R)^3$	equation 6

Most of these equations do not match the response of most thermistors as well as the Steinhart-Hart equation when used over a wide range of temperatures. They may be quite adequate over a narrow temperature range. The advantage of equations 2 and 3 is simplicity and the fact that when you use them, you only need to calibrate the thermistor at two different temperatures instead of three.

The graph below compares using the equation R = A exp (B/T) and the Steinhart-Hart equation to calibrate a Fenwal #FA41J1WC thermistor. The calibration was done at 11°C and 70°C as well as at 40°C when using the Steinhart-Hart equation. Notice that the simpler calibration is very good near the calibration points, but it deviates almost 2°C between these points and is very bad outside the calibration points. The Steinhart-Hart calibration is within one degree over the range 0 – 90°C.



A Comparison of Two Different Thermistor Equations

Program Design Notes

The THERMISTOR program is an Applesoft BASIC program that can determine the count reading from the PDL0 input line and convert that count into a temperature. It also includes a calibration routine that leads the user through the necessary operations to calibrate the probe. The program then calculates the calibration constants and stores them on the disk. Each mode of operation is explained below.

DISPLAY COUNTS

The count reading is actually determined by the machine-language program PDL.CHECK.[5] This program is loaded into memory starting at memory location 768 ($300 in hexadecimal notation) when the THERMISTOR program starts up. Whenever the status of the PDL line is to be checked in the DISPLAY COUNTS mode, the BASIC program executes this statement: `CALL 768`

This statement causes the PDL.CHECK machine-language program to begin operation. The charging time for the capacitor (in counts) is measured. This takes about one second. At the end of the PDL.CHECK program, the count reading is stored in two bytes of memory:

974: low byte 975: high byte

When the PDL.CHECK program is completed, the BASIC program continues. It determines the count reading by checking the two bytes and calculating the counts as a decimal number using this statement:[6]

```
CT = PEEK(974) + 256 * (PEEK(975) - 96))
```

The value of CT is then displayed on the screen as the count reading.

The program will continue displaying counts and temperatures until a key is pressed. It then displays the menu again.

---

[5]The assembly-language source code for this program is in the file PDL.CHECK.SRC on the *Chaos in the Laboratory Programs #1* disk. Refer to *Appendix E* for suggestions on how to examine and modify these machine-language programs.

[6]Because of the way the machine-language program operates, the count total is stored in a complicated way. The number in the low byte represents a number of counts. The number in the high byte starts at 96 and increases by one for every 256 counts registered.

CALIBRATION

In this routine, the count and temperature at three different temperatures are entered into the computer. The program instructs the operator to place the probe and a thermometer into three water baths. In each case, when the thermometer and the probe have reached equilibrium, the operator presses <RETURN>. This causes the computer to take the count reading, which is then used in calibration. The actual temperature is entered by the operator. The three temperature readings and three count readings are then used to determine the constants A, B, and C for the thermistor. These constants are saved in a data file named THERMISTOR.CAL. Whenever the THERMISTOR program begins, this file is read from the disk so that the proper conversion from counts to temperature can be made.

DISPLAY TEMPERATURE

This mode will check and display the count exactly as the DISPLAY COUNTS mode does. It also converts counts to temperatures by using the Steinhart-Hart thermistor equation. The BASIC statements are: `TE = 1 / (A + LOG (CT) * B + (LOG (CT))`^{3} `* C) - 273.3`

TE is the temperature when the count reading is CT. Note that LOG in BASIC used base 10, not base e. The program will continue displaying counts and temperatures until a key is pressed.

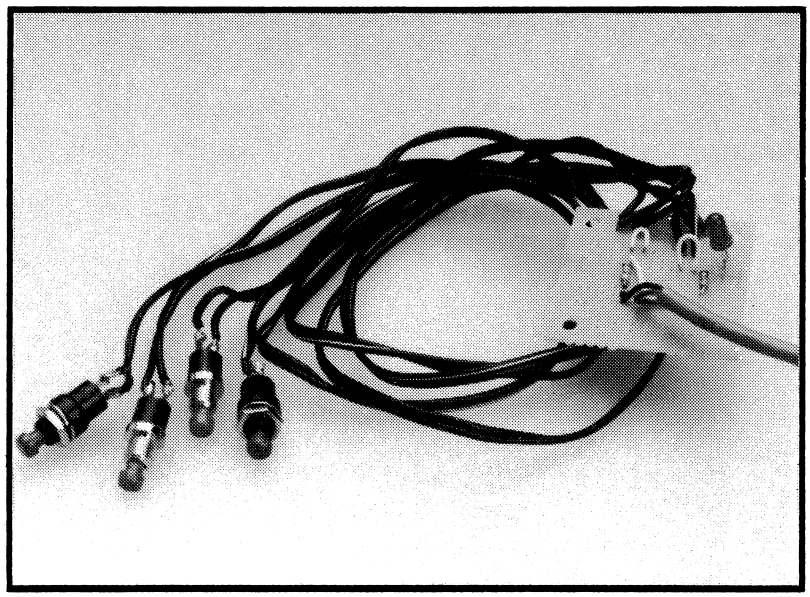
The following variables are used in the THERMISTOR program:

CT(1), CT(2), CT(3)	Count readings at the three calibration temperatures
T(1), T(2), T(3)	Temperatures at the three calibration temperatures
CN	Latest count reading
SO(I, II)	Intermediate values used in calculating calibration constants
A, B, C	Calibration constants
LG	Log_{10} count
TE	Temperature
I	Loop variable for matrix calculations
II	Loop variable for calibration point 1, 2, or 3
J	Delay loop variable
M$	Mode of operation
D$	Control D (used for disk access)
GS	Used to set the Apple IIGS speed to normal
CM	Calibration intermediate value

The 2-probe version of the program (called THERMISTOR.2) is very similar. Reading a second probe that is connected to the PDL1 line is done with the PDL.CHECK program by using a different CALL statement: `CALL 794`

The count total is placed in the same memory location as before. The count can be checked using a PEEK command in BASIC as before. The following variables are used in the THERMISTOR.2 program:

CT(II,P)	Count readings at calibration temperature II for probe P
T(II,P)	Temperatures at calibration temperature II for probe P
CN(P)	Latest count reading
SO(I, II)	Intermediate values used in calculating calibration constants
A(P), B(P), C(P)	Calibration constants for probe P
LG	Log_{10} count
TE	Temperature
I	Loop variable for matrix calculations
II	Loop variable for calibration point 1, 2, or 3
J	Delay loop variable
P	Probe being used
M$	Mode of operation
D$	Control D (used for disk access)
GS	Used to set the Apple IIGS speed to normal
CM	Calibration intermediate value



Project 2

Quiz Show

With this circuit you can have lively, quick-recall contests similar to TV game shows. The core project is to build sets of switches connected to the computer for two or three teams of contestants. The computer can detect which team was the first to press one of its buttons. Project extensions include allowing for more people on a team and a circuit to identify which person on the team pressed his or her button first. There is even a circuit to replace the computer, which uses LEDs to indicate which team pressed one of its buttons first.

Project Requirements

Computer

The core project below will work on an Apple II+, IIe, or IIGS. A special version of the circuit is required for use with an Apple IIc or Laser 128. This special version is described in the "Project Extensions."

Background Knowledge

- Electronics: The core project is very easy. Experience with soldering is recommended. The project extensions are somewhat more complex; specific component placement on the circuit board is not given, so you have to be able to read a simple schematic diagram and decide how to position the components.

- Programming: No programming is required. Some of the project extensions require experience with BASIC programming.

Parts

For connecting the two or three four-person teams to the computer:

- ☐ cable to game port (see *Appendix A*)

For each four-person team:

- ☐ circuit board (Any small circuit board can be used. One possibility is one-half of a Radio Shack #276-159.)
- ☐ (4) push-button switches (normally open) (Radio Shack #275-1556 or #275-618) These should be big and sturdy enough to withstand rough use.
- ☐ 100-Ω, 1/4 watt, carbon resistor (brown, black, brown, gold) (Radio Shack #271-1311)
- ☐ 220-Ω, 1/4 watt, carbon resistor (red, red, brown, gold) (Radio Shack #271-1313)
- ☐ LED (any bright LED such as Radio Shack #276-041 will work. It is best to use the following colors for the LEDs: team #1 – red, team #2 – yellow, team #3 – green).
- ☐ (2) short jumper wires for connections on the circuit board
- ☐ (4) one-meter lengths of two-conductor cable for connecting the switches to the circuit board
- ☐ four meters of three-conductor cable for connecting the circuit boards to the game port cable
- ☐ heat-shrink tubing or electrical tape for insulation of connections

Tools and Supplies

Low-power soldering iron, solder, wire cutters, and wire strippers.

Background Information

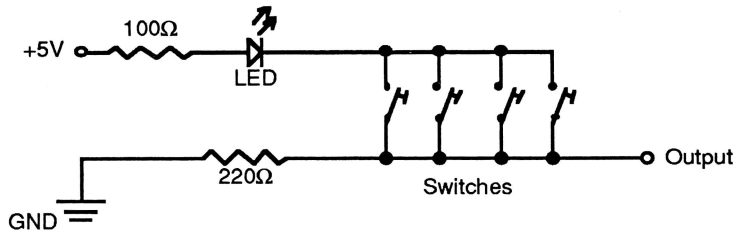
Quick-recall quiz shows have been popular on television for years. Examples include *College Bowl*, *Jeopardy*, and dozens of shows that have faded away as their ratings dropped. Now high school events where teams compete using quick recall are becoming very popular. Contests of this type almost always require an electronic system to determine which team responds first.

There are several ways to do this job electronically. My first quiz show button system (which I still have) was built in the 1960s. It used electromechanical relays and plugged into 120 volt AC electricity. When digital ICs became available, I rebuilt the system using tiny, $1.00 chips to replace the big, heavy, expensive relays.

Now, if you have an Apple II computer, the easiest way to get the job done is to build a simple circuit to connect to the game port and let a computer program help. The hardware for this project consists of a very simple switch circuit that allows you to set up two or three teams of four people each. Whenever any one of the switches on any team is pressed, a signal is sent to the game port of the computer. The program provided on the *Chaos in the Laboratory Programs #1* disk simply checks the status of the game port input lines and indicates on the monitor the team with the first button pressed. Any buttons pressed after the first one are ignored by the program until the master of ceremonies presses a key on the computer indicating that it is time for a new question.

The three teams are identified by colors: red, green, and yellow. We encourage you to build the quiz show system using LEDs that match the team colors.

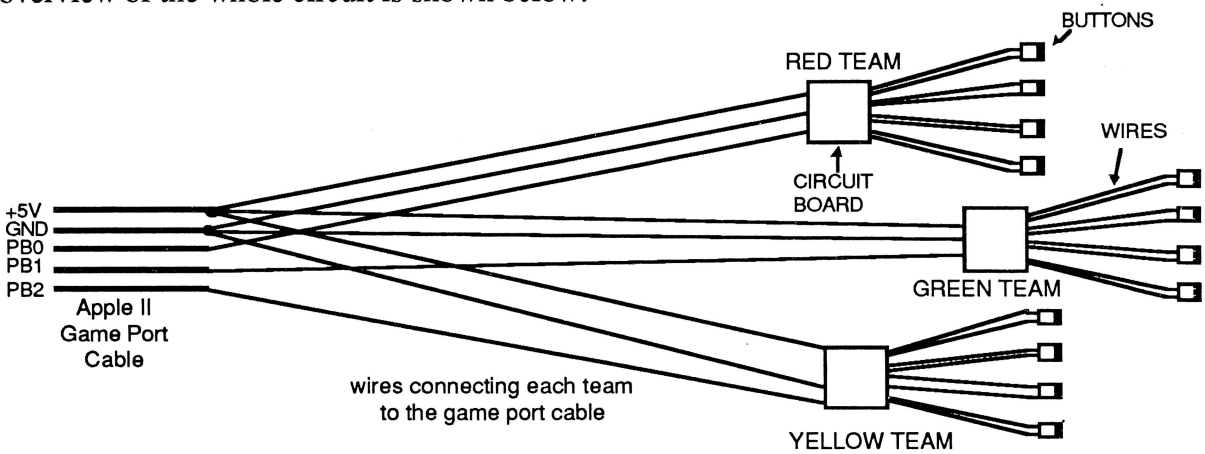
The schematic diagram for a four-person team is shown below:



Four-Person Team Circuit

Essentially, the circuit is just four switches connected together in parallel. Whenever any one of them is pressed, the output pin changes from its normal low voltage to a higher voltage. At the same time, current will flow through the LED causing it to light. The output pin of the team circuit is connected to one of the PB digital inputs on the Apple II game port. PB0 is used for the red team, PB1 for the green team, and PB2 is used for the yellow team. The third (yellow) team is optional. If you want to have just a two-team match, simply do not build a third button system. The resistors in this circuit are to limit current flow through the LED and were selected to make the Quiz Show circuit work with most Apple II-compatible computers (II+, IIe, and IIGS). The Apple IIc, IIc+, and Laser 128 require a slightly different circuit. The circuit for these computers is included as a project extension.

An overview of the whole circuit is shown below:



Quiz Show Circuit Overview

In this diagram, the four switches for each team member are shown very close together. Actually, you can add as much wire as you want between switches on a team. Similarly, you can make the wires as long as you want to separate the teams. The only limitation is the cost of the wire and the added hassle of storing a lot of wire without tangling it.

This project has lots of room for expansion. If you want to have more people on a team, you can simply add more switches. There is no limit to the number of switches per team. In some cases, it may be important to know not just which team pressed a button first, but which person on the team pressed first. A project extension allows you to set up such a circuit. Finally, you may want to use the quiz show button system in a situation where the computer is not available. A project extension shows you how to build a "computer replacement" circuit.

The program QUIZ on the *Chaos in the Laboratory Programs #1* disk is written entirely in Applesoft BASIC. It simply checks the status of the game port and reports a color indicating the team that pressed first. It displays the color in large letters so a large group can see which team gets to answer. You can try the program even before you build the switches if you are using an Apple IIe, IIGS, or IIc computer. Run the program QUIZ. When asked how many teams you will be using, enter 2. The <OPEN APPLE> key on the keyboard will act as a switch for the RED team and the <CLOSED APPLE> key will act as a switch for the GREEN key. Many suggestions for adding to the QUIZ program are given in the "Project Extensions."

Step-by-Step Instructions

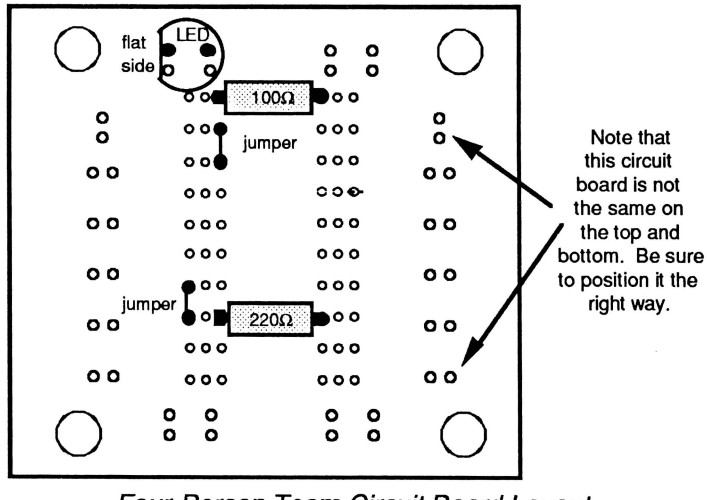
Notes: These instructions assume you are using one-half of a Radio Shack #276-159 Dual IC Circuit Board for mounting the components. If you are using any other circuit board or a prototyping board for assembling this circuit, you will have to lay out the circuit yourself. Use the schematic diagram in the "Background Information" as a wiring guide.

These instructions are for use with an Apple II+, IIe or IIGS. If you plan to use an Apple IIc, IIc+, or Laser 128, you will have to modify this circuit as explained in the "Project Extensions."

☐ 1. **Collect the circuit boards, switches, wire, and components you need.** Using the parts list at the beginning of this chapter as a guide, gather all the parts you need. This circuit is simply a set of switches in parallel that connects to a small circuit board. One switch is required for each team member. There is one circuit board per team. Each circuit board requires one LED and two resistors. You will need some 2-conductor cable to connect the switches to the circuit board. The length of these wires determines how far apart members of one team can be.

A three-conductor cable will connect each team circuit board to the game port cable. The length of these cables determines how far apart the teams can be. One game port cable is required for the entire project.

☐ 2. **Place the components on the circuit board.** Hold the circuit board (one-half of a Radio Shack #276-159) with the copper-plated side down. The components should be placed on the side without the copper plating, with the leads poking through to the copper-plated side. Place the LED on the circuit board as shown below. Use a red LED for the first team. It must be oriented with the flattened side to the left. Now, position the two resistors and two short jumper wires on the circuit board. The "left-right" orientation of the resistors does not matter. Position the body of the resistors down flat against the circuit board.

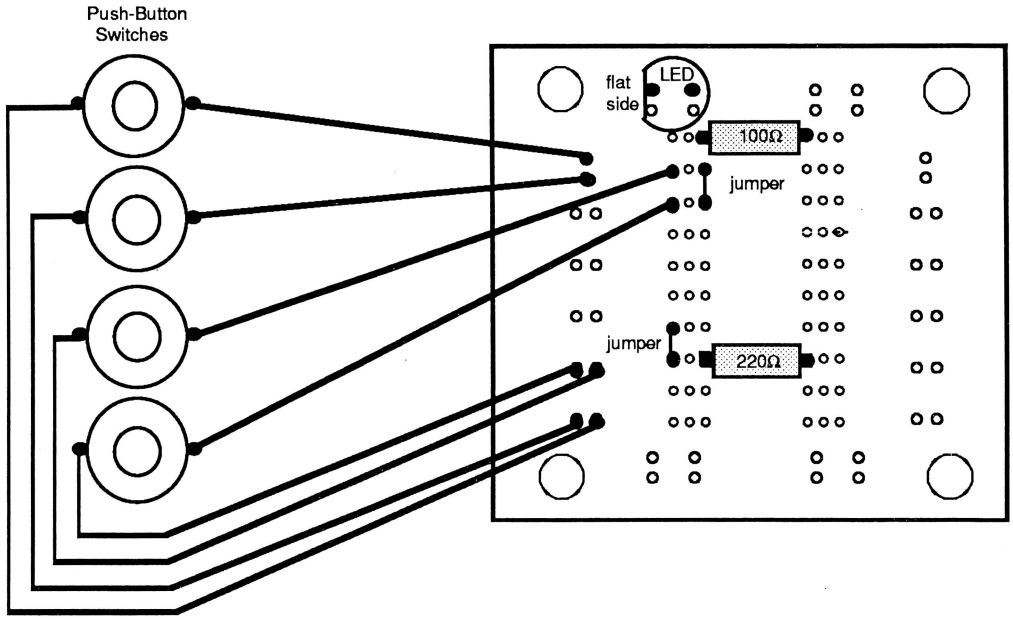


Four-Person Team Circuit Board Layout

☐ 3. **Solder the components to the circuit board.** Turn over the circuit board and solder all of the lead wires to the plated side of the board. When soldering the LED, it is a good idea to use a heat sink on its leads to prevent overheating. After soldering, cut off the excess leads from all of the components.

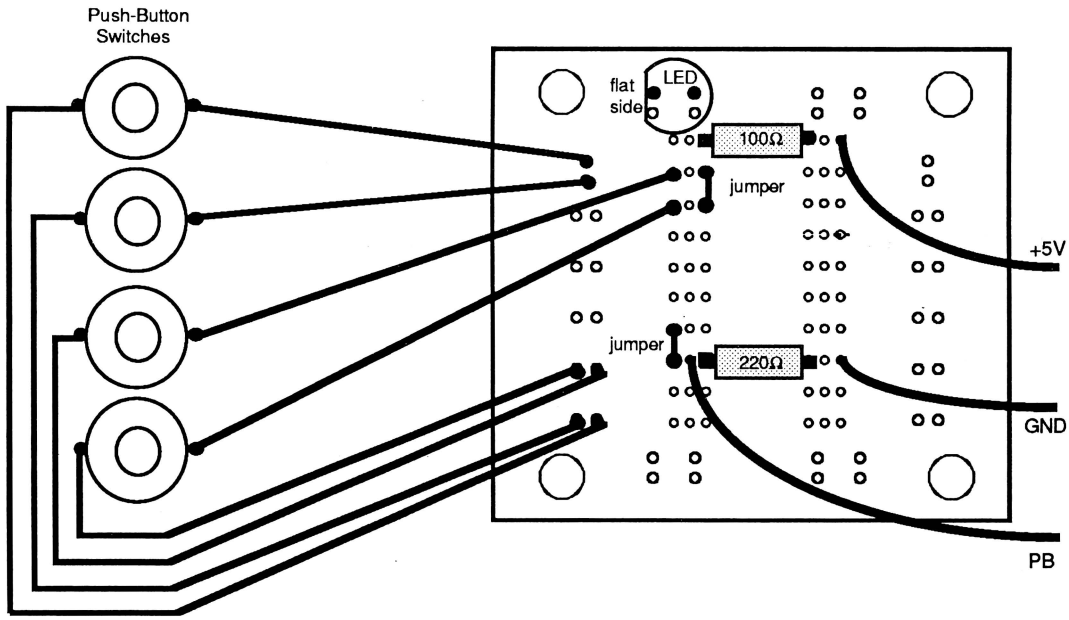
☐ 4. **Solder the 2-conductor cable to the switches.** You will need a two-conductor cable for each push-button switch. These cables will connect the switches to the team circuit board. Strip a few millimeters of insulation off both leads on both ends of all four 2-conductor cables. For each of the four switches, solder one lead of the two-conductor cable to each of the terminals on the switch.

☐ 5. **Solder the 2-conductor cable to the circuit board.** Solder the wires from the switches as shown in the diagram below. Cut off the excess wire. Check for solder bridges or stray strands of wires that could cause short circuits.



☐ 6. **Prepare the 3-conductor cable.** You need to decide how far apart you want the quiz teams to be and make the 3-conductor cables long enough to reach to the computer cable. Two meters should be long enough in most cases. Cut a piece of 3-conductor cable to the length you want for this team. Strip about 1-cm of insulation from the three leads on each end.

☐ 7. **Solder the 3-conductor cable to the circuit board.** Solder the 3-conductor cable to the points labeled +5V, GND, and PB in the diagram below. You will need to know which wire is which on the other end of the cable. If your wire is color coded, make note of the color used for each connection.



☐ 8. **Trim and inspect your circuit.** Trim the wires on the copper-plated side of the circuit board. Look for solder bridges where excess solder has connected two points that should not be connected. Also make sure all the solder connections are filled with solder and are shiny and smooth.

☐ 9. **Prepare the cable to the Apple game port.** Refer to *Appendix A* for information on cables. Use either a 9-pin cable or a 16-pin ribbon cable. Only the following leads will be used: +5V, PB0, PB1, PB2, and GND. Strip a few millimeters of insulation from the ends of these five leads. You might want to label or note the colors of the five wires for future reference.

☐ 10. **Connect the 3-conductor cable to the computer cable.** Solder together the +5V wires from the computer game port cable and the 3-conductor cable to the circuit board. Do the same for the GND wires. The PB wire from the 3-conductor cable should be connected to the PB0 wire if this is the first team circuit, PB1 for the second, and PB2 for the third. Once you are finished with a connection, insulate it with electrical tape temporarily. You will have to remove the insulation from the GND and +5V connections when you add new teams. If you have heat-shrink tubing available, it can be used to permanently insulate the PB connection, since it will not be used for another team.

☐ 11. **Test the circuit.** Turn off the computer. Plug the Quiz Show circuit into the Apple game port. Turn on the computer and run the PB.TESTER program on the *Chaos in the Laboratory Programs #1* disk. With the program running, press each button on the circuit to make sure that the status of the PB0 line changes from "LOW" to "HIGH" when the button is pressed, and that the LED on the circuit board lights up. If you do not get these results, refer to the "Troubleshooting" section below.

☐ 12. **Build circuits for more teams.** You cannot have much of a quiz show with only one team, so the next thing to do is build additional circuits for your competition. Follow the instructions above to build one or two additional Quiz Show circuits. The only changes are:

a) In the step where the 3-conductor cable connects to the computer cable, the PB wire from the circuit board should go to the PB1 lead (for the second team) or PB2 lead (for the third team), instead of PB0.
b) Use a green LED for the second team and a yellow LED for the third team.
c) The +5V and GND leads from the additional teams' circuits are soldered onto the existing +5V and GND connections.
d) When you test the new switches, a different PB line should change status.

When you finish all of the teams, carefully insulate all of the connections using electrical tape.

☐ 13. **Try out the QUIZ program.** The simple program QUIZ on the *Chaos in the Laboratory Programs #1* disk can be used to run the button system during a real contest. Try it out. The program will ask you whether you have two or three teams. It will then indicate that it is ready for the first question. As soon as any button on any team is pressed, it will announce the team, using the color codes RED, YELLOW or GREEN. Only the first press of a button will affect the computer until the program is reset by a press of the <SPACE BAR>. If you have a few people available, try a small practice quiz show session. Make sure that all of the buttons are working.

If you are using an Apple IIGS, you may want to use the GS control panel to turn off keyboard buffering. This will make the <SPACE BAR> function more naturally. Choose the control panel, then KEYBOARD, and then set KEYBOARD BUFFERING to OFF.

☐ 14. **Mount the switches.** The switches are difficult to use without some kind of mounting. Some possibilities that we have used are plastic boxes for electronics projects, 35-mm film canisters, plastic pill bottles (for a hand-held version), or even one end of a tin can.

Troubleshooting

Check the following if your Quiz Show system is not working properly:

If the computer will not start up:

◊ This indicates a short circuit (direct connection between the +5V and the GND leads). Check these leads.

If none of the switches on any team work:

◊ Make sure that you have the +5V and GND wires right. Also, check to see if you have used the correct wires for the PB0, PB1, and PB2 connections.
◊ If you are using a 16-pin game port cable, is it inserted with pin #1 in the right spot? Are all 16 pins inserted into the socket?

If some teams work and the other teams do not:

◊ Check to see if you are using the wrong wire for the particular PB line.
◊ Check the +5V and GND connections to that particular circuit board.
◊ Check the appropriate circuit board for a solder bridge or a loose wire.

If certain switches on a particular team do not work, but others do:

◊ Check the connections to the particular switch.
◊ Check the circuit board for a solder bridge or a loose wire.
◊ Make sure the LED is positioned properly with the flat side oriented as shown in the working diagram.
◊ Disconnect the circuit from the computer's game port and check the resistance across the switch in question using an ohmmeter. It should change from very high (millions of ohms) to almost zero each time the button is pressed. It should return to a very high resistance when the button is released. Bad switches are a frequent problem.

If none of this helps:

◊ Check the circuit board for a solder bridge or a loose wire.
◊ Check the voltage at the PB line. It should be below 0.8 volts when the button is not pressed, and above 2.0 volts when the button is pressed.

Project Extensions

Program Improvements

Add new features to the simple QUIZ program, such as:

◊ Add a question-generation routine. Change the program so that it contains a bank of questions that are randomly selected and displayed on the monitor. The questions could be kept on a disk file or in memory.
◊ Keep score. Store the results from each question and report the latest score after each question.
◊ Add timing. Allow only a specified time period for answering each question. At the end of the time period, the program could display "Time is Up" and sound a beep.
◊ Store the results on disk, either individually or as a tournament.
◊ Keep track of tournament scores, statistics, etc., and display or print the progress.
◊ Use graphics to print the scores in large digits so that many people can read them.
◊ Combine Project 11 (D-to-A Converter) to make the QUIZ program announce the team that pressed the button using digitized sound.

Circuit Modifications

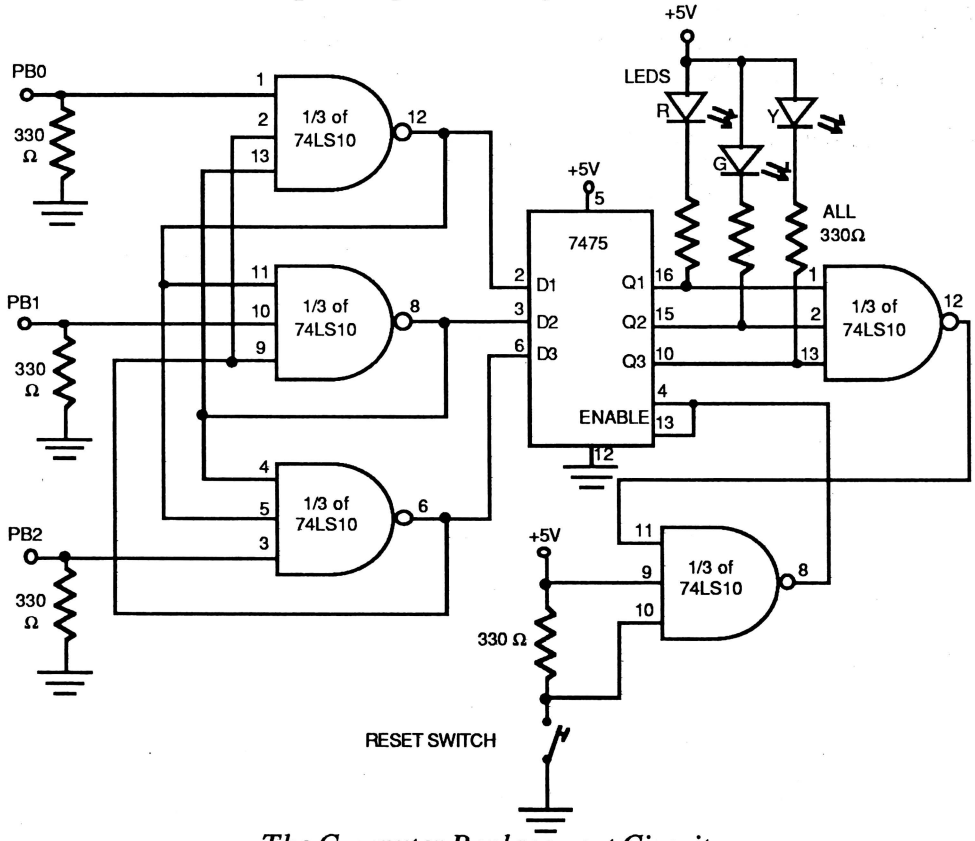
ADD ADDITIONAL SWITCHES FOR A TEAM

This is an easy addition—all that is involved is adding more switches. If you look at the circuit board, you will notice that the copper-plating connects one lead from each of the four switches to the 100-Ω resistor, and the other lead from each switch connects to the output wire. To add more switches, simply connect them to the same places. The switches are all in parallel; any number you want to use can be added.

COMPUTER REPLACEMENT

You can build this circuit to monitor your quiz show button system without a computer. It has an LED for each team that will stay on after the team presses one of its button until the master of ceremonies presses the reset button.

The complete schematic for the computer replacement system is shown below:

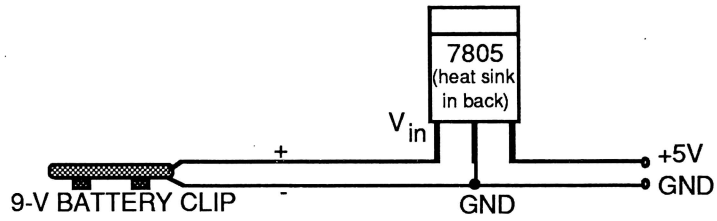


The Computer Replacement Circuit

This circuit requires two 74LS10 ICs and one 7475 IC. You will have to decide on the best way to lay out the components on your circuit board. It is best to use sockets for the ICs instead of soldering them directly to the board. You will have to use jumper wires to make the connections between the pins as shown.

Make this circuit using the same connector for the +5V, GND, PB0, PB1, and PB2 leads as your computer's game port. This will allow you to use your Quiz Show button systems with either the computer or this computer replacement circuit by just plugging it in.

As drawn above, the computer replacement circuit requires a 5-volt power supply. If you want to make the system totally portable, use the circuit below to supply the 5 volts from a 9-volt battery.



5-Volt Power Supply

You may want to examine how the computer replacement circuit works in some detail. It is a good example of a digital logic circuit. It uses two types of *TTL* (transistor-transistor logic) digital ICs. Digital electronics is based on the use of only two voltage levels, either high (above 2 volts) or low (below 0.8 volts). These voltage levels are used to indicate the status of certain conditions in the circuit. The NAND gate used in this circuit (74LS10) has three inputs. The voltage level on the inputs controls the voltage level of the output. If all of the inputs are at high, the output will be low. If any of the inputs are low, the output will be high.[1] The button systems used in this project produce a low voltage except when they are pressed. When the buttons are connected to this circuit and none of the switches are pushed, the switch input of each NAND gate is low. The output of all the NAND gates will be high. Note that the output of each NAND gate is connected to the inputs of the other two NAND gates. These outputs are normally high. When the switch input on one of the NAND gates goes high (when a button is pressed), all of the inputs on that NAND gate will then be high. Since all of its inputs are high, the output of the NAND gate goes low. It also will effectively lock out the other NAND gates, because it feeds a low signal to the other two NAND gate inputs, making it impossible for their outputs to go low.

The other IC in this circuit, the 7475, is called a *transparent latch.* As long as its ENABLE pin is held high, whatever level is present on the D inputs will also appear on the Q outputs. When the ENABLE pin changes to low, whatever voltage level is present on the Q outputs will remain there regardless of changing levels on the D inputs. When the voltage level on the Q outputs is high, the LED has no voltage across it and it does not light. When a Q output goes to low, there is a voltage difference created across the LED and it will glow. The latch locks the system in one status until something happens to change it. That something is the pressing of the reset switch. The reset switch controls the voltage applied to the enable line of the latch, specifying when the status of the Q lines can be changed. See if you can explain how the voltages on the Q pins and the reset line work with the two 74LS10 NAND gates to make the enable line go low when the reset button is pressed.

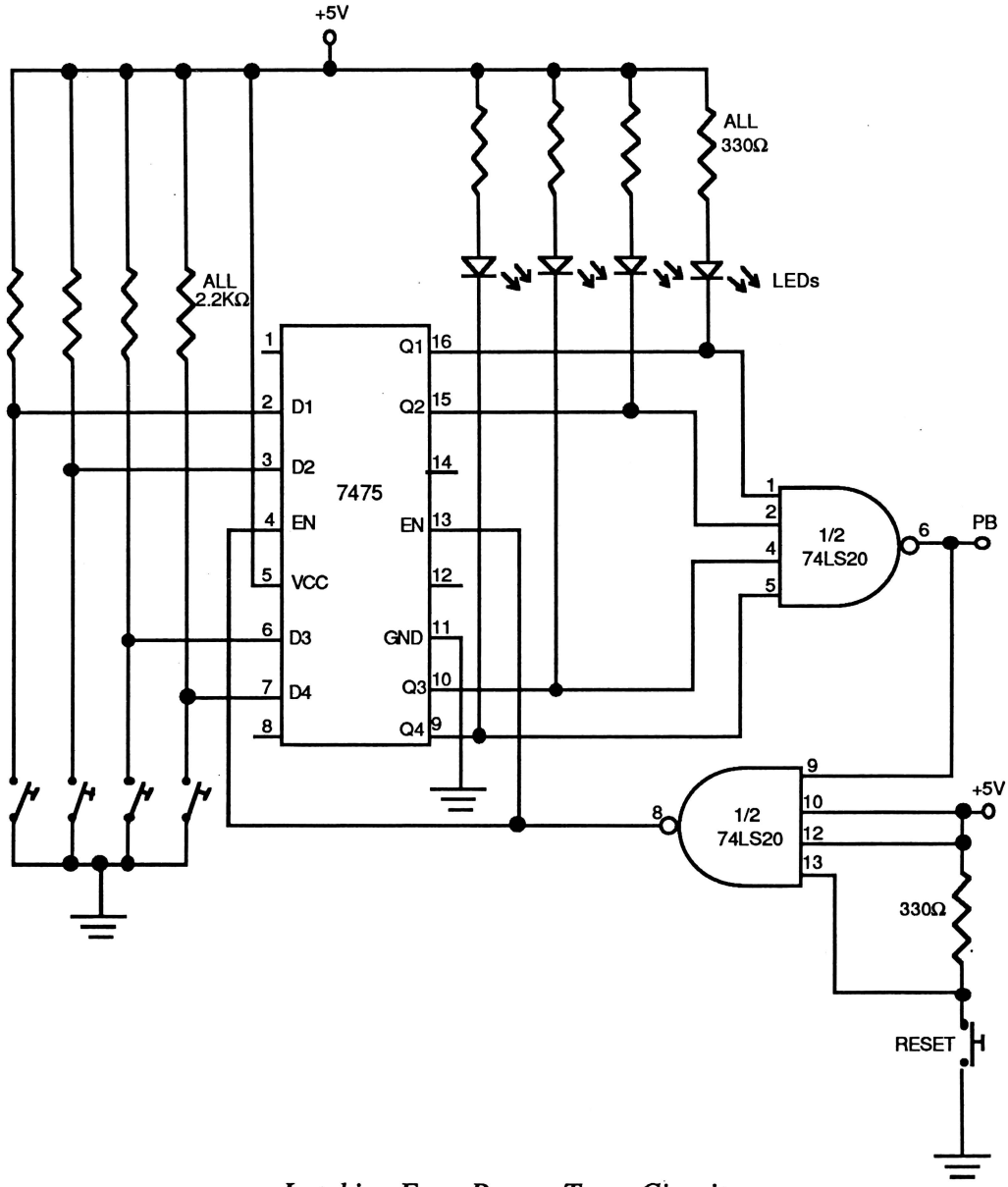
LATCHING 4-PERSON TEAM

You can use this more complicated circuit to replace the simple Quiz Show team circuit. This circuit will *latch* the first person on the team to press his or her button. This means that after one person has pressed his or her button, all other persons on the team are locked out until the reset button is pressed. This makes it possible to see which person on a team pressed his or her button. This circuit will work on all Apple II computers except a IIc, IIc+, or Laser 128.

The complete schematic for the latching 4-person team circuit is shown on the next page. You might notice that this circuit looks a lot like the computer replacement circuit described in the previous project extension. That is because they are basically doing the same thing. Both circuits wait for one of the inputs to go low, and then latch so that the other inputs are disabled until the reset button is pressed. You will have to decide on the best way to set up the circuit on your circuit board. This circuit uses two ICs: a 7475 (the transparent latch used in the computer replacement circuit) and a 74LS20 (a

[1]NAND is short for "NOT AND." One type of logic gate is called an "AND gate." If all of the inputs of an AND gate are high, the output is high; otherwise, the output is low. A NAND gate is the same as an AND gate with the output inverted.

4-input NAND gate). It is best to use sockets for the ICs instead of soldering them directly to the board. You will have to use jumper wires to make the connections between the pins as shown.



Latching Four-Person Team Circuit

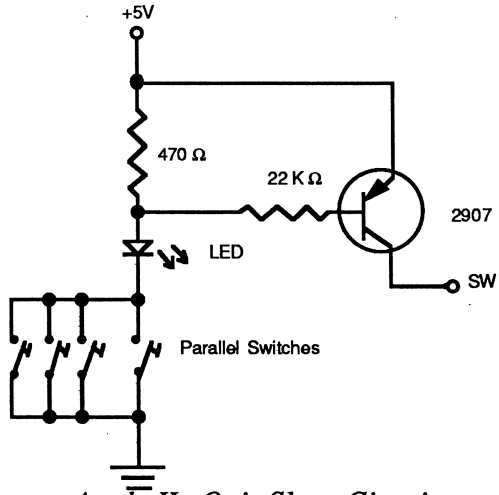
MULTIPLE LATCHING 4-PERSON TEAMS WITH ONE RESET BUTTON

The 4-person latching team described just above has a team reset button which is pressed to clear the LEDs. If you make two or three latching teams, a nice refinement would be to connect all of the teams to a single reset button controlled by the master of ceremonies. Build each latching team circuit as described above, but connect pin #13 of the 74LS20 on each circuit to the same switch. This switch should cause pin #13 on each team to go to a low voltage when it is pressed.

CIRCUIT CHANGES FOR USE WITH APPLE IIc, IIc+ AND LASER 128

The internal differences in the game port of the Apple IIc, IIc+ and Laser 128 computers will not allow the use of the standard Quiz Show circuit described in the core project. The addition of a transistor and another resistor is necessary. The complete schematic for the Quiz Show circuit to be

connected to the Apple IIc, IIc+ or Laser 128 is shown below. You can add as many switches in parallel as you want with this circuit.



Apple IIc Quiz Show Circuit

REPLACE THE PUSH-BUTTONS WITH TOUCH-SENSITIVE SWITCHES

Electronic hobby magazines and books often include circuits for switches which are touch sensitive. They use the resistance or capacitance of your body to switch on or off. You may want to experiment with circuits of this type.

Supplemental Information

Program Design Notes

The QUIZ program is a fairly simple Applesoft BASIC program. The major sections of the program are outlined below to make it easier for you to work with and modify:

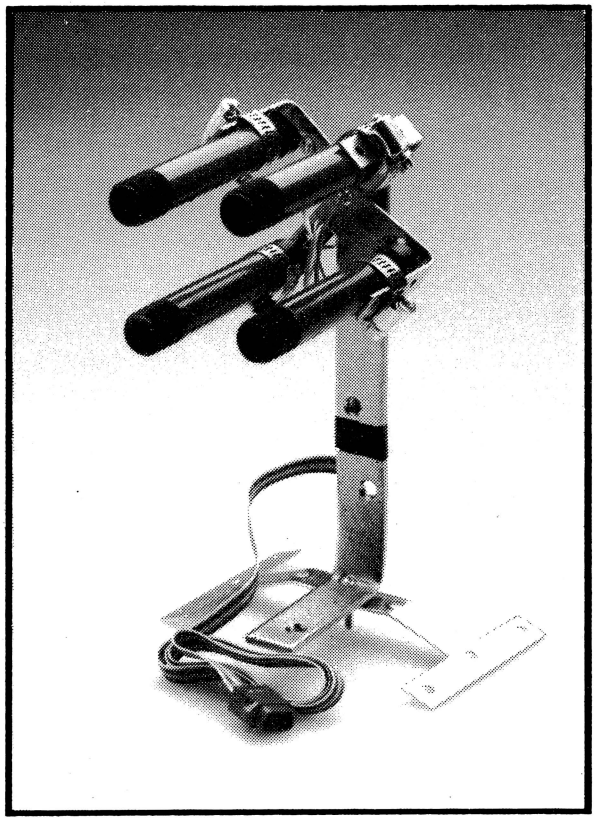
Lines 5 to 90: Initialize the program and set the values of the constants. This section also asks the user whether there will be 2 or 3 teams.

Lines 100 to 235: This is the main loop of the program that waits for a switch to be pressed.

Lines 240 to 280: Displays a color identifying the team that pressed the button and then prompts the user to press a key to go on to the next question. The program will not continue if one of the buttons is still pressed.

Lines 600 to 665: This is the subroutine that works with the LP.1200 machine-language routine to draw the color of the team that pressed the switch on the screen.

The LP.1200 machine-language program is used to quickly draw large letters on the low-resolution graphics screen. You may want to use it in your own programs. Use this program as an example. To use it from Applesoft BASIC, first BLOAD it into memory. Then within the program, poke the ASCII code of the letter to be printed into memory location 4611. The position of the start of the letters is POKED into 4612 (horizontal) and 4613 (vertical). Use a CALL 4608 statement to actually draw the character.



Project 3

Computer Vision

by Satoshi Kino

In this project you will build a "camera" for the Apple II+, IIe, or IIGS that allows the computer to identify objects and patterns. The computer identifies objects by comparing them to *templates*, which are sets of data about objects previously digitized. The sets of data are based on how much light the object reflects. Although the Computer Vision system works only under controlled environments (lighting and positioning of the object should always be the same), it is surprisingly accurate. With this vision system you can have your computer identify everything from apples and oranges to your friends' faces. This is a good project for students interested in artificial intelligence.

Project Requirements

Computer

This project uses the four PDL input lines that are available on Apple II+, IIe or IIGS computers. It cannot be done using other models of Apple II computers.

Background Knowledge

- General: Building the apparatus requires minimal construction skills. Some of the extensions may require some experience with woodworking or metalworking.
- Electronics: The electrical circuit in this project is very simple. Some previous experience with soldering is desirable.
- Programming: No programming experience is required for the core project. Some of the extensions require experience with BASIC programming.

Parts

It is helpful to read through the "Step-by-Step Instructions" once before obtaining the parts. After you understand how each part is to be used, it will be much easier for you to make substitutions and modifications to your parts list. The total cost of the parts is about $20 if you have to buy everything new. More likely, you will be able to find some spare parts around your home or lab to reduce the cost.

- ☐ (4) photoresistors (Radio Shack #276-1657 is a pack of five photoresistors, which are referred to as photocells)
- ☐ cable to Apple game port (see *Appendix A*)

The parts listed below can be purchased at local hardware stores. The metal brace, corner braces, and t-plates must have pre-drilled holes for proper construction. All measurements are approximate and any of the parts may be substituted by equivalent materials. For example, Erector® Sets may be used to construct the Computer Vision system.

- ☐ (1) metal brace 12" x 1" (You will need to make one right-angle bend in this piece, so make sure it is not too stiff to be bent.)
- ☐ (4) corner braces 2 1/2" x 5/8"
- ☐ (2) T-plates 4" x 4" (A block of wood can be substituted)
- ☐ (2) bolts, nuts, and washers. Bolts should be reasonably long (1") and fit into T-plates, metal brace, and corner braces.
- ☐ (4) metal, plastic, or sturdy paper tubes about five inches long. The inside diameter of the tube should be equal or slightly larger than the diameter of the photoresistors. The tubes can often be found in the plumbing section of a hardware store. We used 1/2" tubes.
- ☐ (4) hose clamps with adjustable diameter from 1/2" to 1 1/4"
- ☐ flat black spray paint
- ☐ glue (use epoxy, silicon, or a hot glue gun)
- ☐ plastic electrical tape or heat-shrink tubing for insulating connections
- ☐ extra wire (optional) to connect between the photoresistors and the game port cable

Tools and Supplies

A soldering iron, solder, wire cutters, wire strippers, screwdriver, and pliers are all required. A vise may be necessary to bend some of the metal pieces.

Background Information

When we see objects, we are actually interpreting the light waves being reflected off the object. The brain processes the light patterns and compares the object it sees to objects it has seen before. Although it is impossible to model the recognition power of the brain, it is possible to create a simple computer vision system which mimics the biological eye.

In this computer vision system, photoresistors are used as "eyes."[1] Photoresistors change their electrical resistance as the intensity of the light striking them changes. As more light hits the photoresistor, the resistance of the photoresistor decreases.

The Apple II has four resistive inputs to which four photoresistors can be attached. The inputs, PDL0, PDL1, PDL2, and PDL3, can report values from 0 to 255 which correspond to the amount of light striking the photoresistor. By running the following test program, the computer gives a continuous reading of the light levels of the four photoresistors (run PDL.TESTER on the *Chaos in the Laboratory Programs #1* disk):

```
10 PRINT PDL(0), PDL(1), PDL(2), PDL(3), PDL(4)
20 GOTO 10
```

If you connect four photoresistors to the computer and point them all in the same general direction, you have a crude camera. If you place a dark object such as a black piece of paper in front of the camera, the values reported by the computer increase. Conversely, a light colored object will register smaller values. Even a slight change will affect the readings. The photoresistor cannot tell the difference between colors, only the relative reflectivity of an object. In other words, the photoresistor is sensitive only to light intensity and not wavelength, unlike the human eye, which is sensitive to both. The following chart shows the reflectivity values of various colored sheets of construction paper:

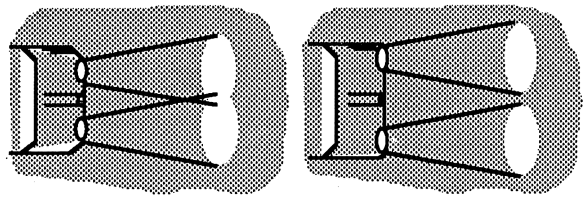
Color	PDL Value
Yellow	53
Light blue	69
Grey	71
Orange	78
Dark orange	85
Green	117
Red	130
Purple	133
Blue	150
Brown	159
Black	255

Typical Counts from the Photoresistors

The photoresistors can tell the difference between the different colored papers because the light level, distance to the construction paper, and material out of which the paper was made were all kept constant. For unvarying conditions, even a slight color change can be detected by the photoresistors. Using a chart similar to that shown above, the computer can say that it is looking at a light blue piece of paper if it receives a value of 69 or a green piece of paper if the value is 117.

The Computer Vision system is set up so that each of the four photoresistors will cover only one quadrant (one fourth) of an object. The photoresistors are mounted inside short tubes. The tubes limit the field of view of each photoresistor. The Computer Vision system functions much like car headlights. If the headlights are spread too far apart, objects appearing in the middle of the road will not be illuminated. On the other hand, if the headlights are too close together, the lights are not being used efficiently. A median spacing must be maintained so that the objects are illuminated without any gaps and with minimum overlap.

[1]Photoresistors are sometimes referred to as *photocells*. Radio Shack uses that term for the photoresistors it sells.



The photoresistors on the Computer Vision system follow the same principle. If the photoresistors are placed too close (Figure 1 below), each of the photoresistors will be covering part of the other photoresistor's region. However, if the photoresistors are positioned too far apart (Figure 2 below), a gap will occur in the center and important portions of the scanned object might be missed. Figure 3 shows optimal overlap of the four photoresistors. The tubes are close enough to eliminate any gaps while minimizing the overlap region.[2]

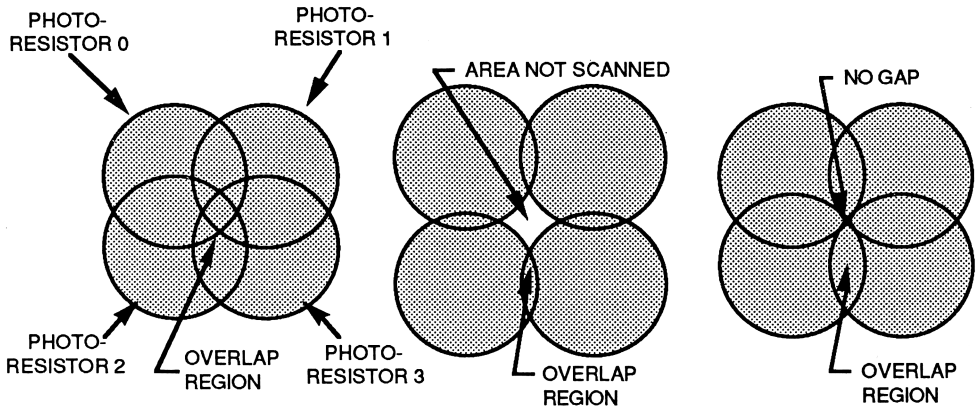
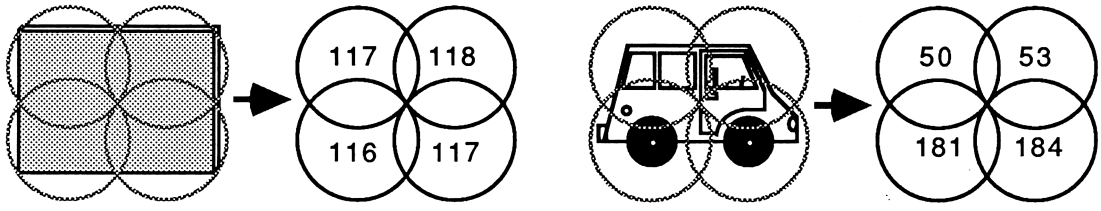


FIG. 1 TUBES TOO CLOSE *FIG.2 TUBES SPREAD TOO FAR* *FIG 3. OPTIMAL OVERLAP*

Given an object that is much more complex than a sheet of construction paper, the computer will have more trouble recognizing the object. For example, if we scan a model car that has an average reflectivity level of 117, the computer will have a difficult time determining the difference between the car and the green piece of paper. However, the fact that the four photoresistors are positioned in a square pattern allows the computer to extract more information on which to base its decision.



GREEN PIECE OF PAPER
AVERAGE REFLECTIVITY: 117

MODEL CAR
AVERAGE REFLECTIVITY: 117

The figure above shows what values the computer might receive when scanning a green piece of paper and a model car. For the green paper, the values of each of the photoresistors are quite close to the mean value of 117. This would indicate that the object is uniform in reflectivity. Although the model car has the same average reflectivity value as the green piece of paper, the top two photoresistors report much smaller values than the lower two photoresistors. The dark tires in the bottom half of the model car are responsible for the larger values in the lower photoresistors. This contrast in the image allows the computer to tell the difference between the two objects.

---

[2]For more information on how the Computer Vision system works and how to calculate optimal overlap, refer to the "Supplemental Information" section.

In the computer's memory, the information for the various images are stored in an array similar to the following chart:

Objects in Computer's Memory:

Object Name	PDL0	PDL1	PDL2	PDL3
Green paper	117	118	116	117
Blue paper	152	151	148	149
Model car	50	53	181	184
Model plane	73	40	55	68

The first column contains the name of the object. The second column contains the value of the first photoresistor, the third column contains the values of the second photoresistor, and so on. Each row represents one object.

The computer receives the following information and is asked to identify the unknown object:

Unknown Object to be Identified:

PDL0	PDL1	PDL2	PDL3
48	54	179	183

Again, the table contains similar information. The first column is the value of the first photoresistor, the second column is the value of the second photoresistor, and so on. For the computer to identify this object, it must determine which object in the computer's memory most closely matches the unknown object. A total sum of the absolute difference between the unknown and each of the objects in the computer's memory is calculated. For example, for the green paper, the difference value would be 117-48 + 118-54 + 179-116 + 183-117 = 262. (See the "Supplemental Information" section for more on how to compute the summation of difference). Continuing with this example, the difference value for each of the objects in this case would be:

Object Name	Diff. Value
Green Paper	262
Blue Paper	266
Model Car	6
Model Plane	278

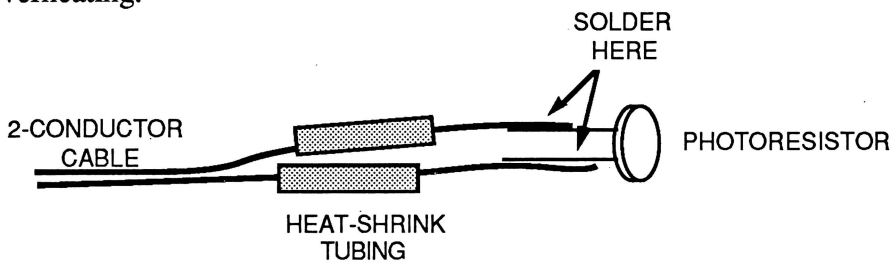
The computer calculates the sum of the absolute difference and looks for the smallest value. The object with the smallest difference value is named by the program as the "unknown object." In this case, the unknown object is a model car because it has the smallest absolute difference value. The next closest object is the green paper with a difference value of 262. Since there are no other objects with a small difference value, the computer chooses the model car as the correct object.

The resolution of the Computer Vision system limits its ability to recognize specific distinguishing features such as a car has two tires, the car has windows, an airplane has two wings, or an orange is round. It can only correctly identify objects it has seen before based on its reflectivity of light. For objects it has not seen before, it makes a rough classification based on the amount of light reflected in each of the four quadrants. Despite its limitations, this algorithm is surprisingly accurate in identifying objects it has seen before. Objects like apples, oranges, and even people's faces can be identified successfully.

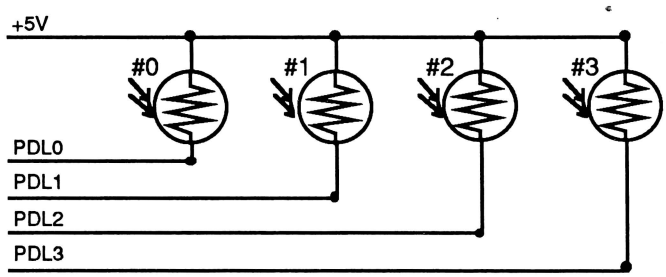
Step-by-Step Instructions

☐ 1. **Spray paint each of the tube's interiors flat black.** Hold the tube up to the nozzle of a can of flat black spray paint. Spray paint to coat the inside of the tube. Leave the tube standing upright to allow excess paint to drain out of the tube. Allow the paint to dry and apply another coat of paint. Be sure that the entire inside of the tube has been painted. The paint minimizes the reflection of light off the inside of the tube. Let the paint dry while you work on the electrical part of the project.

☐ 2. **Attach lead wires to each of the photoresistors.** Two-conductor cable will lead from each of the photoresistors to the game port cable. These cables should be about a foot (30 cm) long.[3] Solder a 2-conductor cable to each of the photoresistors. Use heat-shrink tubing or plastic electrical tape to insulate these connections. Use a heat sink on the photoresistor lead while soldering to prevent damage due to overheating.



☐ 3. **Attach the photoresistor cables to the game port cable.** Refer to *Appendix A* and identify the following five leads: +5, PDL0, PDL1, PDL2, PDL3. If you plan to use heat-shrink tubing, be sure to slide it on the wires before beginning to solder. Twist together one of the leads of each 2-conductor cable and the +5V lead from the game port.[4] Solder these five wires together and insulate the connection with plastic electrical tape. Solder the other lead of each cable to one of the PDL leads. Heat-shrink tubing can be used to insulate these connections. The electrical circuit of the completed circuit is shown below:



The Computer Vision Circuit

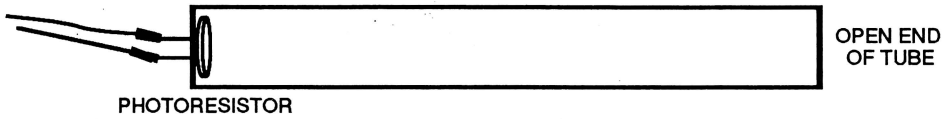
☐ 4. **Test the circuit.** Turn off the computer and plug in the photoresistor circuit. Turn on the computer. Run the PDL.TESTER program on the *Chaos in the Laboratory Programs #1* disk. The first column displays the values for PDL0, the second column is PDL1, and so on. If all of the photoresistors are exposed to light, the four values should be between 0 and 255. The values should vary as the light shining on the photoresistors changes; zero is the brightest and 255 is the darkest value. If the photoresistor circuit does not work properly, refer to the "Troubleshooting" section.

---

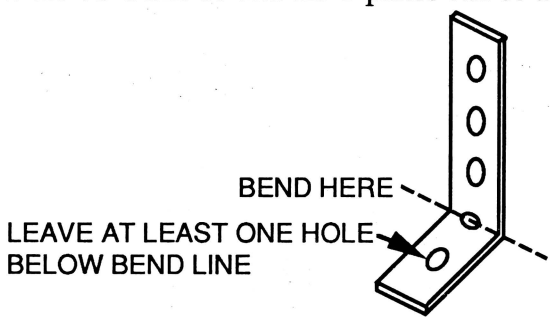
[3]The length of cable to use here depends on your application—how far away from the computer will you be using your Computer Vision System? The game port cable itself will extend from the computer for a few feet. Make these wires longer if you would like to add extra distance; it will not cause any electrical problems.

[4]There is no "+" side to a photoresistor, so you do not have to worry about which of the two leads to use.

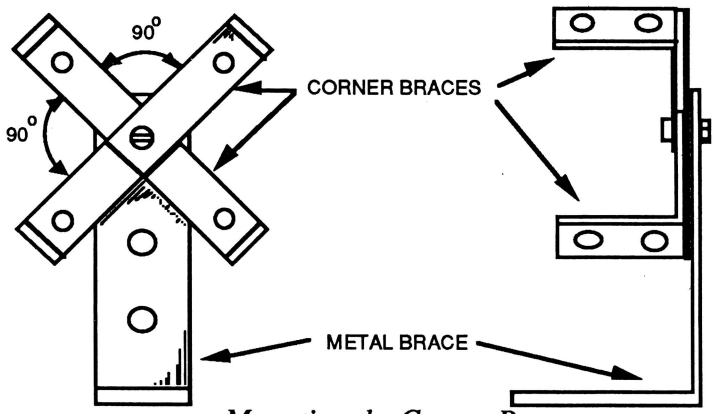
☐ 5. **When the paint on the tubes is dry, attach the photoresistors to the end of the tubes.** A glue such as epoxy works well for holding the photoresistors. Hot glue guns can be used, and have the advantage that you can hold the photoresistor in place while the glue hardens quickly. Position the photoresistors such that the back end of the photoresistor is flush with the end of the tube. Be sure that the photoresistors are not tilted. If the photoresistors do not fit snugly, try wrapping some masking tape around them. Allow the glue to harden. Black electrical tape should then be used to seal the ends so that no light leaks through.



☐ 6. **Bend the metal brace into an "L" shape.** Bend the brace 90 degrees. It is easiest to bend the metal at a point where a hole has been drilled. You may have to use a vise to bend it. In general, the metal brace should be bent at about one third of the total length of the metal piece. Be sure to leave at least one hole below the bend line so that the T-plates can be attached to the metal brace.



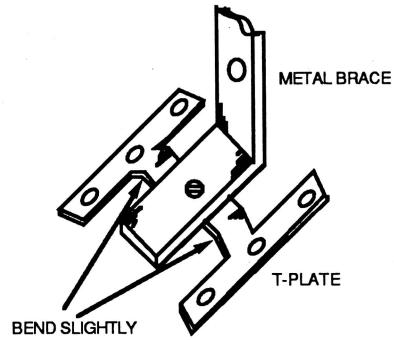
☐ 7. **Attach the four corner braces to the top hole in the metal brace.** Pass a 1" long bolt through the holes in each of the corner braces and attach it to the top of the metal brace. Place a washer and a nut on the bolt. Arrange the four corner braces in a square pattern as shown below and tighten the bolt:



Mounting the Corner Braces

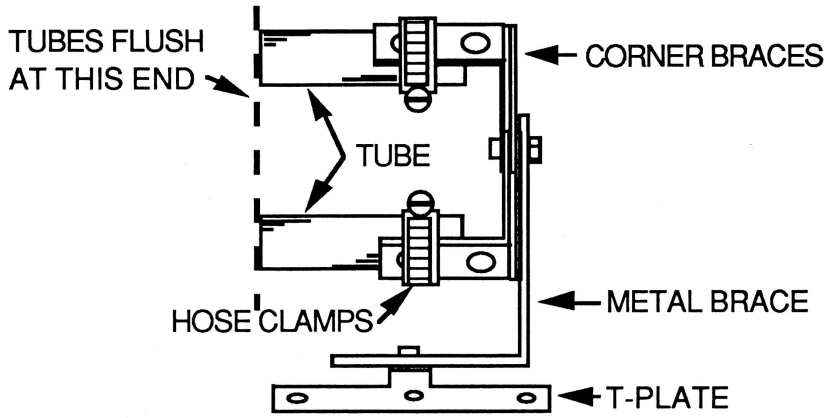
Front View *Side View*

☐ 8. **Attach the T-plates to the bottom of the metal brace so that the structure will stand freely.** Pass the 1" bolt through one of the holes in the brace and attach the T-plates. Place a washer and a nut on the bolt. Arrange the two T-plates in a "H" pattern and tighten the bolt. Bend the T-plates so that the structure will stand freely.



A block of wood may be used as a stand instead. If you are choosing this route, attach the structure to the block with wood screws.

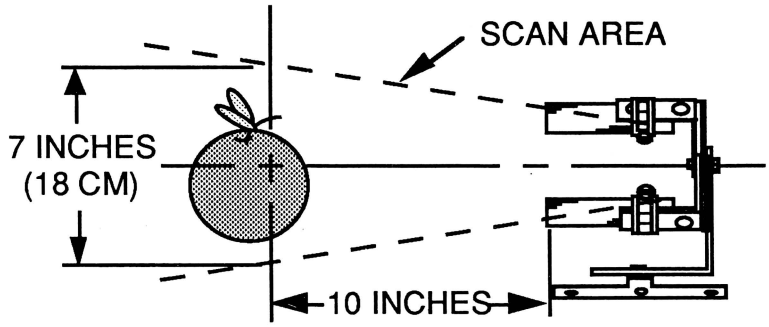
☐ 9. **Attach the four tubes to the corner braces using the hose clamps.** Place the tubes on the inside of the corner irons and use hose clamps to hold them in place. Arrange the tubes such that they are parallel to each other and the ends of the tubes are flush as shown below:



☐ 10. **Digitize objects and see if the computer can recognize them.** A good way to make sure the hardware is still working is to connect the Computer Vision system to the computer and run the program PDL.TESTER again. PDL.TESTER displays the count from each of the four photoresistors. A different set of values should be displayed when a different object is placed in front of the Computer Vision system. Make sure you have fairly bright light in the room. If the system seems to function properly, you are ready to move on.

For the initial Computer Vision experiment, use objects that are quite different in color, luster, shape, and size (avoid objects like spoons and forks). Apples, oranges, bananas, and carrots are a good selection of objects to try. Objects should be less than 7 inches (18 cm) square in size and should be placed about 10 inches from the Computer Vision system (see figure below).[5] Make sure the area is well lit and the lighting is constant. Changes in the lighting of the scan area will cause problems. For best results, use a non-glossy background that registers a value near 255. Make sure the objects are always positioned at the same distance from the Computer Vision system and have the same orientation. Keep in mind that the lighting conditions must be consistent; be sure not to cast shadows while data is being taken. While setting up the experiment, it is a good idea to run the PDL.TESTER program and make sure there are significant changes in the counts reported by the four sensors as you move various objects in place. If the count reported by a sensor is always 255, you do not have enough light (or there is a wiring problem).

[5]These figures assume that you have used the parts suggested in these instructions. If you have used different tubes or tube spacing, the proper position for placing your objects and the size of the scan region will be different. The "Supplemental Information" section has a discussion of how to calculate the scan area.



Run the VISION program on the *Chaos in the Laboratory Programs #1* disk. Choose option number 1 - CREATE OBJECT TEMPLATES, to start a database of objects for the computer to use. The computer will display the object's number and will prompt you for the object's name. Enter the object's name and press <RETURN> when the object is ready to be digitized. Continue digitizing until finished or memory is filled (up to twenty objects can be saved in memory).

Exit to main menu and select option 2 - IDENTIFY OBJECTS to identify an object. The current status will be displayed, including the number of objects in the computer's memory and the image averaging value. Press <RETURN> to start digitizing an object. When the computer is finished digitizing and computing, it will come up with the closest template fit to the object it sees. It will also display a list of objects that are close to the unknown object. Press <RETURN> to continue.

Using the VISION program, digitize some objects and see if the computer correctly identifies them. How accurate is the computer? Does it always choose the correct object? What factors seem to influence the computer's choice the most? What happens if you show the computer something it hasn't seen before?

Troubleshooting

Run the PDL.TESTER program. The first column displays the values for PDL0, second column displays the values for PDL1, and so on.

If the PDL values are not the same under the same lighting conditions

Unless you are using very high quality photoresistors, the PDL readings will not be exactly the same. As long as the values are relatively close, the circuit is functioning properly. Also, fluctuations of ±4 in the photoresistor values is to be expected under constant lighting conditions.

If one or more of the PDL values is 255

Check for one of the following problems:

- The cable to the game port is not plugged in correctly—check it. If you are using a 16-pin connector, make sure it is oriented the right way.
- The tube is plugged or covered—check to make sure nothing is covering the tubes.
- The photoresistors need more light—use the Computer Vision system in a brighter area. Try using bright lights to illuminate your objects.
- The circuit is not making a good connection—check solder joints and the connection to the computer.
- The photoresistor may have been damaged due to overheating during soldering—replace them.
- The photoresistor is not seated properly. If the photoresistor is crooked in the tube, it will not receive the light well. Dislodge the photoresistor and glue it back in its place. Be careful not to damage the photoresistor.

If one or more of the PDL values is 0

Check for one of the following problems:

- Short circuit in wiring—recheck wiring to see that no wires are crossed.
- Too much light hitting photoresistor—dim the lights or move to a darker area.

If the PDL value does not change when the light level changes or works only over a very narrow range

Check the following:

- The photoresistor may have been damaged during soldering or the photoresistors are old—check to see that the photoresistor is functioning properly. Replace if necessary.

Project Extensions

Additional Experiments

◊ Identifying people's faces

Set up the Computer Vision system so that you can digitize people's faces. It does a surprisingly good job at this. Unlike objects like cups and boxes, you will probably have trouble getting people to stay in the same position. The best way to do this is to make a box with an oval-shaped hole on one side (like the box described in the Controlled Environment project extension described later in this chapter).

◊ Security

Based on your success of identifying people's faces, is the Computer Vision system reliable as a security system? What happens if a person gets a hair cut or a tan? See if you can find some aspect of a person that is distinctly different, yet does not change drastically over time.

Perhaps a more conventional security system would be a motion detector. Place the Computer Vision system in a corner of a room so that it is able to scan the whole room. Have the computer sound an alarm when a change is detected. If there is a window in the room, what happens when the sun moves and changes the lighting conditions of the room? Perhaps multiple images of the room during the course of a day would be able to compensate for light changes.

◊ Quality control

The most practical application of computer vision systems is quality control. Manufacturing companies often use vision systems to spot defects in parts rolling past on a conveyor belt. In automated factories, the object is usually placed in the same orientation and thus it is easy for the computer to visually isolate the object and identify defects.

Try scanning a series of similar objects for defects. You might want to modify the program so that it will notify you if an object has an absolute difference value outside a given tolerance. As an example, set up the computer to identify a defective plastic fork out of a box of plastic forks.

Some factory conveyor belts will stop so that the camera can "capture" the image and then advance to the next object and stop again. However, in some cases, the conveyor belt keeps moving. Devise a method (hardware or software) that will be able to "freeze" moving objects.

◊ Navigation system on an vehicle

Allowing a vehicle or a robot to maneuver without the aid of a human is a very difficult task. The computer must be able to recognize and navigate around objects while staying on the road. Try

placing the Computer Vision system on a computer controlled car (for example, Project 11 in *How to Build a Better Mousetrap*) and have it follow a black line drawn on the floor. Modify the program and hardware so that the car movements are determined by what the Computer Vision system reports. For a first attempt, try having the car navigate by following a black line drawn on the floor. Is the Computer Vision system able to distinguish objects and areas of the room well enough so that it can maneuver without bumping into things? What kind of modifications must be made to the vision hardware? To the car? Are other types of sensors required for reliable navigation?

Program Improvements

The programs provided on the *Chaos in the Laboratory Programs #1* disk are written totally in Applesoft BASIC and do not use any machine-language routines. The programs are easily modified. Refer to the "Program Design Notes" and "Memory Structure" under "Supplemental Information" for programming details.

Here are some changes that you might want to make to the VISION program:

◊ Add the PDL.TESTER program to the VISION program as a test mode. This would save time in testing the hardware before beginning an experiment.

◊ The computer always chooses the closest match, even if it is not a very good match. Add an error tolerance that would allow the computer to determine if the choice was "good enough."

◊ Since identification of objects is based on the amount of light reflected, a change in lighting conditions will cause the computer to choose the wrong object. Modify the program so that it can identify objects even if the lighting changes. One very simple method is using ratios of the light intensities between the photoresistors. By dividing the values in each photoresistor with the value in other photoresistors, a ratio can be calculated.

◊ The resistance of a photoresistor is roughly inversely proportional to the intensity of the light striking the photoresistor. That is, the quantity 1/count varies in a nearly linear way as the light level changes. Perhaps the VISION program should use the quantity 1/count in its calculations for matching objects. Now it simply uses the count. Change the program and see whether you think the system works better.

◊ The VISION program now uses a simple sum of the difference formula to determine the best match of objects. You might try changing the program to use a different method of calculating the differences. One example would be to use the sum of the squares of the differences (similar to a standard deviation calculation).

◊ Change the program to increase the number of objects it can hold in memory (right now, the limit is twenty). Devise an algorithm that speeds up the computer's search time. How efficient is a disk-based database?

◊ Add an option to save and load the digitized data for future use.

◊ One of the hottest areas of research in the artificial intelligence world is neural networks. Based on the biological brain, neural networks have the ability to actually "learn" patterns, like a pet dog or horse can, by being repeatedly trained. They also are able to recognize patterns that are heavily distorted by noise and static.

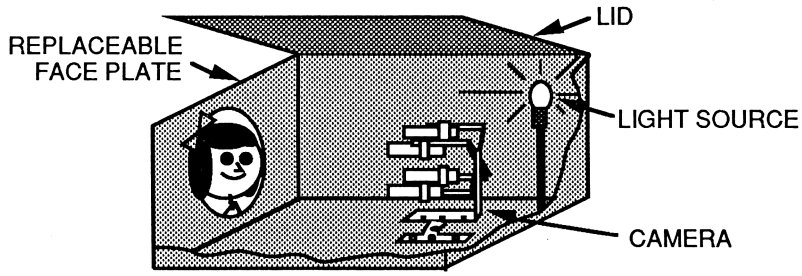
Try modifying your Computer Vision system to use neural networks to identify objects. Although they are rather complex to implement, it would be interesting to see if the results are better. The August 1989 issue of *BYTE* magazine has a good introductory article on neural networks. Also,

the monthly magazine *Artificial Intelligence* is a good source for artificial intelligence applications in vision systems, including neural networks.

Hardware Modifications

◊ Controlled environment

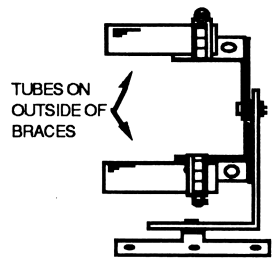
Construct a light-proof box that is immune to outside lighting changes. Make the box large enough so that the Computer Vision system, a light source, and the object to be digitized can fit (see below). For best results, paint the inside flat black and make the box light proof. Try modifying the box so that people's faces can be digitized also. One suggestion is to make the face plate removeable so that a "solid" plate can be inserted in its place. This way, you will be able to digitize other objects without a large hole behind it.



A Controlled-Environment Box in Use

◊ Change the scan area

The original Computer Vision system can only scan an object less than 7 inches (18 cm) square. A quick way to increase the scan area is to move the tubes to the outside of the brackets as shown below. If you are using the tubes and hardware suggested in the "Step-by-Step Instructions," this will increase the scan area to about 10 inches (25 cm) square.



Computer Vision System Modified for a Larger Scan Area

◊ Increase resolution of camera

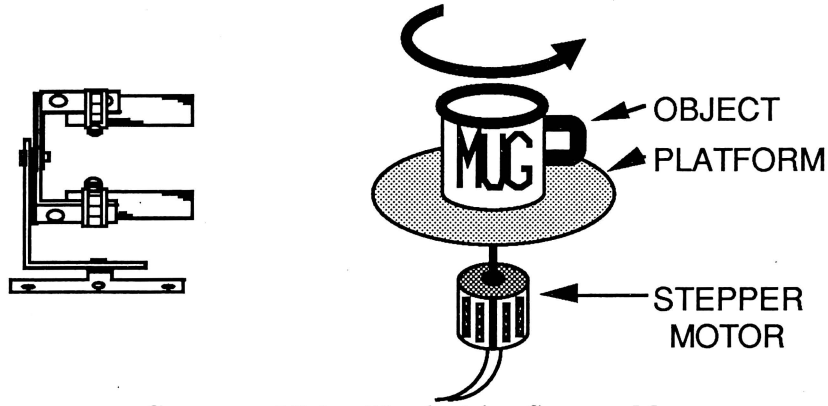
Modify the electronics so that more photoresistors can be added. An analog multiplexer (such as the ADC0816) could be used to increase the resolution of the Computer Vision system. The article by Mark J. Robillard in *Radio-Electronics* and the book *The Robot Builder's Bonanza* by Gordon McComb discuss electronic methods to increase the number of photosensors.

◊ Try experimenting with higher quality photoresistors and other photosensors. Better photoresistors will yield more precise readings. A photodiode or a phototransistor can produce more stable and reproducible results.

◊ Scan using a stepper motor

Add a rotating platform that allows the computer to scan all sides of an object. Use a stepper motor from Project 11 of this book to rotate the platform. Have the computer rotate the platform in 45°

increments. This will give eight readings of an object and will allow the computer to identify objects regardless of which side is facing the Computer Vision system.



Computer Vision Used with a Stepper Motor

Supplemental Information

Memory Structure and Calculation of Summation of Difference

The objects in the computer's memory are stored in a two-dimensional array of real values called S(). The array dimensions are four by twenty. The array can store the four photoresistor count readings for twenty objects.

There is a corresponding character array of dimensions one by twenty that stores the name of the object. Array N$() stores the "permanent" object names. Array P$() is a temporary copy of N$(). Since the name array must be sorted, a temporary copy is needed to maintain the original order of the array N$().

	PDL0	PDL1	PDL2	PDL3
1	50	53	181	184
2	73	40	55	68
3	234	32	247	154
4	26	73	7	23
5	85	12	34	45
6	95	56	123	12
7	23	82	53	23
8	64	[illegible]	5	[illegible]
[illegible]	[illegible]	123	[illegible]	34
17	[illegible]	236	46	34
18	74	26	24	18
19				
20				

Array S(): Contains average count values of twenty objects.

	Name
1	Car
2	Plane
3	Cup
4	Can
5	Bag
6	Dish
7	Box
8	Bill
9	[illegible]
[illegible]	[illegible]
18	Craig
19	
20	

Array N$(): Contains name of corresponding objects.

The unknown object to be identified is stored in a two-dimension array called F(). Array F() has dimensions of one by four and contains the average values of the four photoresistors read in from the Computer Vision system.

PDL0	PDL1	PDL2	PDL3
70	41	57	66

Array F(): Contains average count values for unknown object.

The computer calculates the difference between each corresponding reading of the unknown object and each object in the computer's memory. It then takes the absolute value of these differences and adds them up to get a "sum of the difference" total. This value is a measure of how well the unknown and the object being compared match. A mathematical formula for what the program does is:

$$\sum_{t=1}^{4} |S(X,t) - F(t)|$$

This is the formula for comparing the Xth template with the unknown object F(). The index t steps through each of the four photoresistors counts.

For example, using the sample data in the table above, the summation of difference value for the first object, called CAR, would be calculated as follows:

$$|(50-70)| + |(53-41)| + |(181-57)| + |(184-66)| =$$
$$|-20| + |12| + |124| + |118| =$$
$$20 + 12 + 124 + 118 = 274$$

The second object, called PLANE, would be calculated similarly:

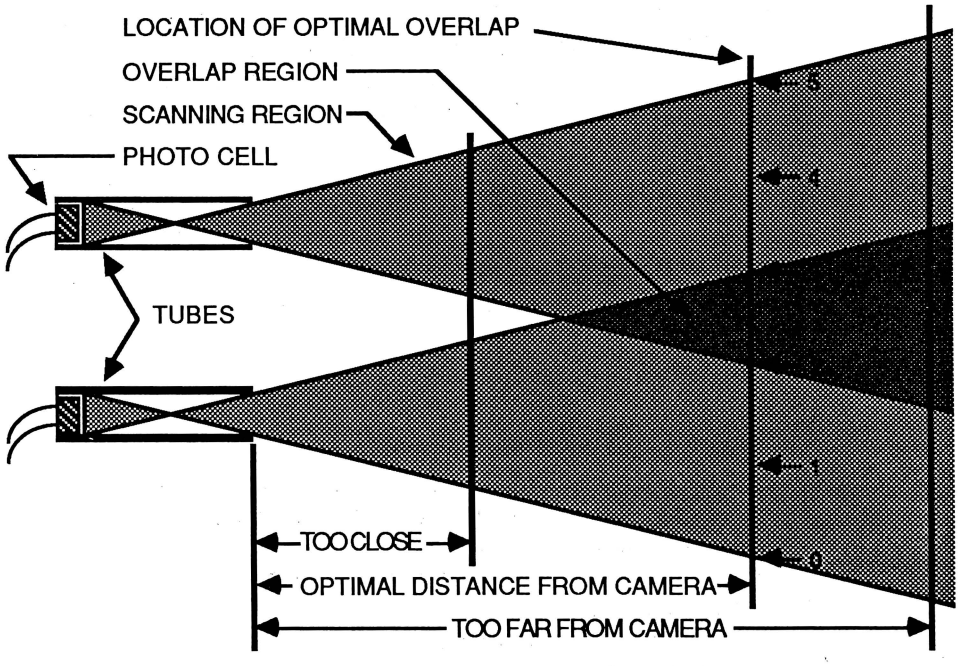
$$|(73-70)| + |(40-41)| + |(55-57)| + |(68-66)| =$$
$$|3| + |-1| + |-2| + |2| =$$
$$3 + 1 + 2 + 2 = 8$$

This "sum of the difference" total is calculated for all of the objects. The values are stored in a single-dimension array of twenty elements called CC(). The array CC() is sorted in ascending order. The corresponding elements in array P$() are also swapped. Since the array CC() holds the difference values between the unknown object and the objects in the computer's memory, the object with the smallest absolute difference is the closest match. The object at the top of the array is the program's pick for the object.

How to Calculate the Scan Area

The Computer Vision system (if built with parts with the same dimensions as the parts list) can only scan an object about seven inches (18 cm) square while maintaining optimal overlap. If you wanted to scan an object significantly larger or smaller, the system would not effectively digitize the object. Although a larger object could be scanned by placing it farther away, there would be so much overlap that the computer would not be able to distinguish any contrast in the object (see figure below). Also, placing a object closer to scan a smaller object would not work too well, because gaps would occur in the scanning area.

To scan an object of different size, the tubes must be moved. Increasing the distance between the tubes will allow the Computer Vision system to scan larger objects. Decreasing the distance between the tubes will allow the system to scan smaller objects. After moving the tubes, the distance at which optimal overlap occurs must be determined. The best way to calculate this is to make a scale drawing of the tube location and draw the field of view of each photoresistor, as shown below:



The key points the drawing should show are the inside diameter of the tube, length of the tube, and the distance between the tubes. Be sure to take into account the thickness of the photoresistor. The optimal overlap occurs where the overlap region is one-fifth of the total scanning area. For example, if the region to be scanned is ten inches square, the overlap must be about two inches. The position at which this occurs is where objects should be placed to be scanned most effectively.

Program Design Notes

Here is a section-by-section explanation of the VISION program:

Lines 10 to 240: Initialization
This section initializes all variables and arrays used. Line 240 sets the image averaging value (W), the number of times the object is scanned. The larger this number is, the more accurate the data will be. However, if the value is too large, it will take longer to acquire the image. A value of 50 is a good compromise.

Lines 500 to 540: Main line
This section branches to the subroutine requested by the user.

Lines 1000 to 1090: Main menu
This section displays the main menu and asks the user for a selection.

Lines 3000 to 3330: Digitize objects
This section digitizes objects. Lines 3000 to 3015 initializes the variables and displays the header information. The object counter A is incremented and the name of the object is requested from the user in lines 3040 to 3140. The values for the object are read in and stored in a two-dimensional array S() in lines 3160 to 3290. In lines 3310 to 3330, the user is asked if another object is to be digitized.

Lines 10000 to 12210: Identify objects
This section is where the actual identification process occurs. In the lines 10000 to 10580, the variables are initialized and the current status of the program is displayed. The number of objects in memory and image averaging value are displayed. The image acquisition occurs in lines 10700 to 10740.

In order for the object to be identified, the data must be sorted in ascending order based on the calculated difference values. Thus, in lines 10860 to 10960, the name array is transferred to temporary arrays. By doing this, the integrity of the actual name array is preserved. The summation of difference formula is in lines 11110 to 11170. The array containing the difference values is sorted in lines 12000 to 12060.

In lines 12080 to 12210 the object that the computer "thinks" is the closest match is displayed. The object with the smallest difference is the object that most closely resembles the unknown object. It also prints out the other possible candidates.

The following variables are used in the VISION program:

A	Number of digitized objects in memory
B0,B1,B2,B3	Variables used for storing sum of PDL values
C$	User input string
CC()	Array of sum of PDL differences between template and object
F()	Array of PDL readings from object to be identified
I,J	Loop control variables
MT	Number of array elements for digitized objects
N$()	Array for template names
P$()	Temporary array for template names
Q	Menu choice variable
Q$	User input string
S()	Two dimensional array for templates
T	Loop control variable
TEMP	Temporary variable
TEMP$	Temporary string variable
V	Loop control variable
W	Image averaging value (initially 50)

For Further Information on Computer Vision Systems

The books and magazines listed below are good sources of information about computer vision and artificial intelligence.

◊ Electronics extensions and modifications:

Robillard, Mark J. "Simple Solid-State Vision." *Radio-Electronics*, November 1986, pp. 74-75.

Robillard, Mark J. "Experimental Robot Vision." *Radio-Electronics*, December 1986, pp. 81-83.

McComb, Gordon. *The Robot Builder's Bonanza: 99 Inexpensive Robotics Projects*. Blue Ridge Summit, Pennsylvania: TAB Books Inc., 1987.

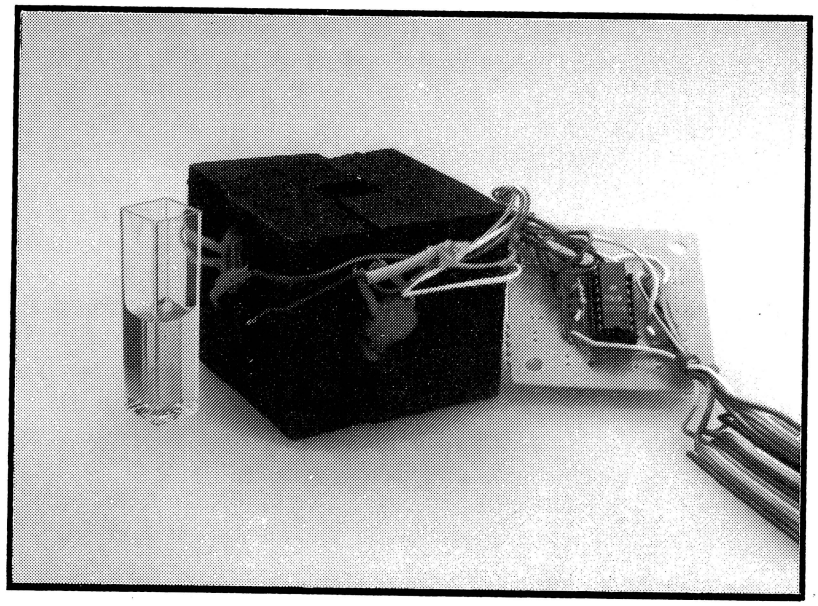
Iovine, John. "Interfacing Commodore's User Port, Part 6: Building a Digital Camera, Part 2." *Commodore Magazine*, January 1988, pp. 56-61.

◊ For more information on advanced image processing methods:

Dawson, Benjamin M. "Introduction to Image Processing Algorithms." *BYTE*, March 1987, pp. 169-186.

Blankenship, John. *Apple IIe Robotic Arm Projects*. New Jersey: Prentice-Hall, 1985.

Frenjel, Jr., Louis E. *Crash Course in Artificial Intelligence and Expert Systems*. Indianapolis: Howard W. Sams & Co., 1987.



Project 4

Colorimeter

by J. Roger Bacon

In this project you will build a device that allows you to study how solutions absorb light. This is a standard method used by chemists to determine the concentration of solutions. We will use light of one particular color at a time. An instrument that does this job is called a *colorimeter*. A wooden block is used to hold the sample while three different colored LED light sources and three photoresistors measure how much light of each color is transmitted. The block is designed to have a square hole in the center that is the correct size for a disposable plastic *cuvette* (a flat-sided container for holding solutions). The amount of light transmitted can be mathematically related to the concentration of the solution.

Commercial colorimeters or spectrophotometers are among the most frequently used instruments for chemical analysis. Your colorimeter will allow you to carry out simple colorimetric methods of analysis.

Project Requirements

Computer

This project uses the annunciator output lines, which are available only on the Apple II+, IIe and IIGS computers. It cannot be done using other Apple II computers.

Background Knowledge

- Electronics: This project requires experience with soldering and using a multimeter for troubleshooting circuits.
- Woodworking: The ability to use a drill press and table saw or a radial arm saw is necessary. **You must understand the hazards of these machines. If you are not experienced with the use of these tools, do not attempt this project without the help of an experienced woodworker.**
- Programming: The program supplied will allow calibration of the device and operation of the unit for chemical analysis. Some of the project extensions require experience with BASIC programming.

Parts

- ☐ panel nails, 1.5 inches long (these nails have grooves around them)
- ☐ knot-free, soft wood shelving (3/4" thick and 1 ft long)
- ☐ flat black paint
- ☐ red, yellow, and green LEDs (Radio Shack #276-041, #276-021 and #276-022)
- ☐ (3) photocells (Radio Shack #276-1657, Mouser #524-611-J4-805, etc.)
- ☐ circuit board (one half of a Radio Shack #276-159)
- ☐ 7404 Hex Inverter Integrated Circuit (Radio Shack #276-1802)
- ☐ 14-pin IC socket (Radio Shack #276-1999)
- ☐ 47-Ω resistor, carbon, 1/4 watt (yellow, violet, black, brown)
- ☐ 16-connector cable to Apple's internal game port (refer to *Appendix A* for information)
- ☐ epoxy putty (Other glues can be used, but epoxy putty has two characteristics that make it work well for this project: It can be applied as a putty and it is opaque, so it will not let light leak into the colorimeter.)
- ☐ square plastic disposable cuvettes (These are rectangular-shaped containers for the solutions to be tested. One source is part #24102-12 from Hach Company, 1-800-227-4224. These are 1" x 1" cuvettes sold in a pack of 12 for about $20. Cuvettes are also available from Cole-Parmer Instrument Company, 7425 North Oakpark Avenue, Chicago, IL 60648-9930, (800) 323-4340, part #6343-25. These are 1/2" by 1/2" and are sold in a package of 200 for about $20. Another source is Sargent Welch Scientific. The best source of just a few cuvettes may be your local chemistry department.
- ☐ miscellaneous wire for making connections between the photoresistors, LEDs, and circuit board.

Tools and Supplies

Low-power soldering iron, solder, wire strippers, drill press, and a table saw (or radial arm saw). It is possible to use a hand saw and a hand-held electric drill; however, the result will probably be better if power tools are used. For testing and mixing solutions you will need food coloring, beakers or glasses, a graduated cylinder (50 ml), and water. For practical use, you will probably want to get some reagents specifically designed for testing specific chemicals, such as chlorine, dissolved oxygen, various pollutants, etc.

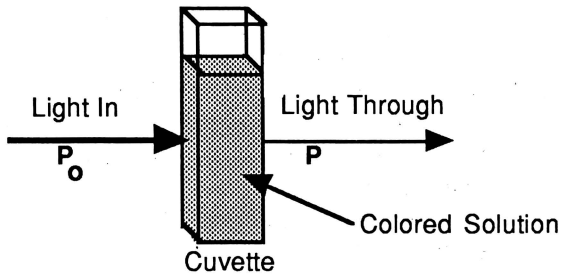
Background Information

When we observe a colored solution we are doing a crude form of colorimetric analysis. We are all aware of the fact that more dilute solutions are less colored than more concentrated solutions of the same material. Colorimetric analysis simply takes the observation away from the limitations of our eyes and uses instrumentation to make these light measurements very accurately. The Beer-Lambert

Law (usually just called Beer's Law) is used in chemical analysis and states that the absorbance of light (A) by a colored solution is proportional to the path length (l) and the solution concentration (c). Mathematically, the law is expressed as A = a l c. The proportionality constant (a) is called *molar absorptivity* if the solution concentration is in moles/liter.[1]

The radiant power of light going through a solution can be measured with various electronic devices, including phototubes, photomultiplier tubes, phototransistors, and photoresistors. In this project, we will use photoresistors. The resistance of the photoresistor is approximately inversely proportional to the intensity of light striking it. The standard Apple II PDL( ) command is used to read the resistance. The computer software takes the raw game port readings and converts them into values that are proportional to light intensity.[2]

Solutions to be studied are placed in a clear container that has a fixed light path length. This container is called a *cuvette*. First, a blank solution (one containing none of the material being measured) is placed in the cuvette and the radiant power transmitted through it is measured. This intensity is referred to as P_O. The sample is then placed in the same cuvette and that radiant power is measured and given the symbol P. The transmittance (T) is calculated as the ratio of P/P_O. The absorbance (A) is calculated from the transmittance information using the equation $A = -\log_{10}(T)$.[3]

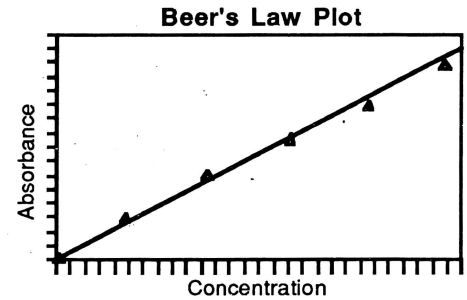


Transmittance = P/P_O

Absorbance = $-\log_{10}(P/P_O)$

Photochemical Measurements

A plot of absorbance vs. concentration for a series of known concentrations can be used to prepare a calibration curve. The Beer-Lambert law states that a graph of this data should produce a straight-line graph with a y-intercept of zero. The slope of the line is equal to the molar absorptivity (a). A typical graph is shown here.



A Typical Beer's Law Graph

Commercial instrumentation for doing colorimetric or spectrophotometric analysis is relatively expensive ($1,000 to $50,000). The main factor in the high cost is that these instruments contain a *monochromator*. This is a device that uses either a prism or a diffraction grating to separate white light into its various wavelengths. Only a single, narrow band of wavelengths comes out of the end of the monochromator. This light shines through the solution, and the absorbance at one particular wavelength of light can be studied. The wavelength of this light can be precisely set, and in many instruments the wavelength can be scanned over a range of wavelengths. A graph of absorbance at various wavelengths (an absorption spectrum) can be made.

---

[1]The proportionality constant is called *absorptivity* if the solution concentration is in grams/liter.

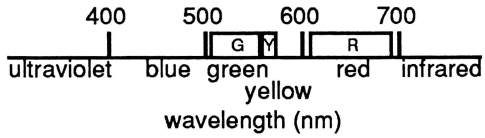
[2]Refer to the "Supplemental Information" section of this project for more information on how the resistance readings from the photoresistor are converted to light intensities.

[3] If per cent transmission is used, the equation for absorbance is: $A = 2 - \log_{10}(\%T)$.

For routine colorimetric analysis, it is usually sufficient to use a few fixed wavelength bands rather than scan the entire spectrum. Getting data on the light absorption of a solution at one or two wavelengths is usually all that is needed. Our colorimeter will use three different colors of LEDs as light sources: red, yellow, and green. We can select the best color of LED to use for studying a particular solution. The LEDs emit light in a band of wavelength centered at approximately these wavelengths:

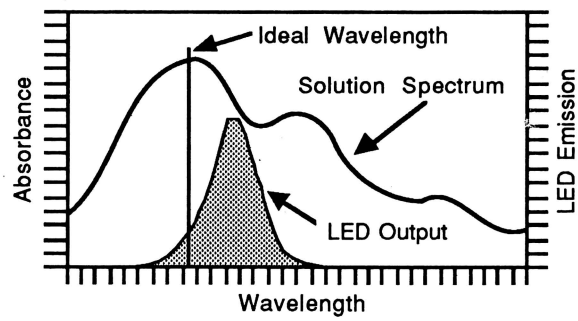
Green: 565 nanometers
Yellow: 585 nanometers
Red: 655 nanometers

The diagram below shows where these colors fall on the visible light spectrum.



The Electromagnetic Spectrum

To decide which LED to use, you need to think about what causes color in a solution. A red solution is red because it transmits red light and absorbs most other wavelengths of light. Using the light from the red LED to study a red solution would not be useful, since no matter how concentrated the solution was, it would never absorb much red light. For studying a red solution, the green LED is probably the best choice, since it emits light that has a much shorter wavelength than red light. The yellow LED could also be used. For a blue solution or a green solution, the red LED should be used. The diagram below shows a complete absorption spectrum for a hypothetical solution. If we had an instrument that could measure absorption at any wavelength, we would pick the wavelength of maximum absorbance. With our system, we choose the LED that produces light nearest the wavelength of maximum absorption.



Once the colorimeter is constructed and the best color of LED is selected, how do we do chemical analysis with it? The procedure typically goes something like this:

First you calibrate the colorimeter by using a *blank* solution (usually just water). This allows the system to set the P_O values for each LED-photoresistor pair. You then take measurements of the absorbance of two or more known concentrations of solution. From this information, you plot a graph of absorbance vs. concentration of the solution. Next you measure the absorbance of an unknown solution. By locating the absorbance value on the absorbance vs. concentration graph, you can then read off the concentration of the unknown solution.

There are a number of interesting topics to investigate using the colorimeter. Environmental chemistry and pollution studies are often done in this way. For example, chemical packets are sold that can be added to water samples of a specified volume to produce color in the solution to indicate the presence of a particular chemical. One example mentioned in the "Project Extensions" uses reagent packages from Hach Chemical Company to measure the amount of chlorine in water. The solution turns pink if chlorine is present, and the absorbance of the solution (when measured using a green LED) varies

proportionally with the concentration. After making some chlorine solutions of known concentration, you can make a calibration graph and determine the concentration of chlorine in pools or drinking water. Similar reagent packets are sold for determining dozens of different chemicals in water supplies.

Step-by-Step Instructions

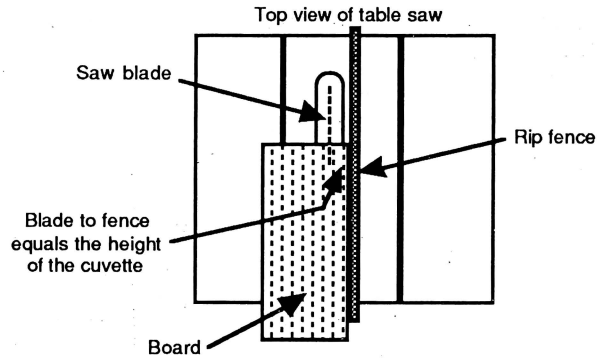
The LED colorimeter is constructed from a small piece of shelving board. Pine or spruce works well, since these woods are easy to cut or drill, and they don't tend to split. Before you start cutting, you need to have your cuvette. Some of the plastic disposable ones that we used have exactly 1/2" by 1/2" outside dimensions; others are somewhat more than one inch wide.[4] Any square, clear, smooth glass or plastic container of about this size would be suitable. Wavy glass such as cheap dropping bottles or containers that are round will give less precise results.

We recommend that you skim through all of these instructions briefly and take a look at the diagrams to get an overview of the project before you begin.

Part One - Woodworking:

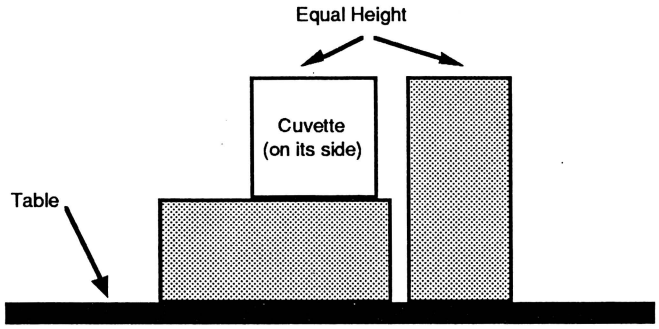
☐ **1. Cut the board to the correct width.** The board should be the same width as the height of the cuvette. Measure one of your cuvettes and cut the whole length of the board to this dimension.

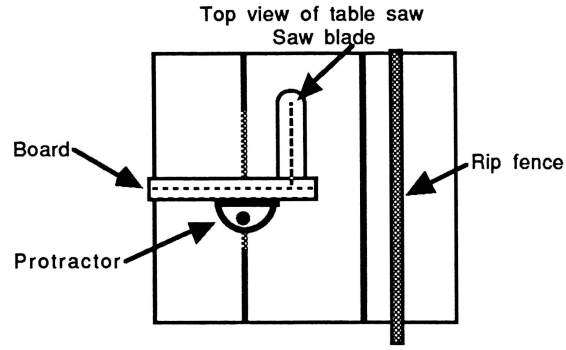
> Note: Cutting very short or very long pieces of wood can be dangerous, so it is best to start with a board that is between 1 and 3 feet long. If possible, get help with these cuts from a shop teacher or a carpenter.



☐ **2. Cut four lengths of the board to be used for the sides.** The length of each piece should be equal to the outside width of the cuvette plus the thickness of the board. The figure below shows how to determine the length of each piece of wood. Set the sliding protractor on the table saw at 90 degrees and cut four pieces off one end of the board. When laid side by side the pieces should be the same, and if you lay the cuvette on one piece, the height should be equal to the length of the other pieces.

[4]Throughout this book we have tried to use metric units. This project uses English units (inches and feet) because the cuvettes and lumber in the United States are usually sold in these units.

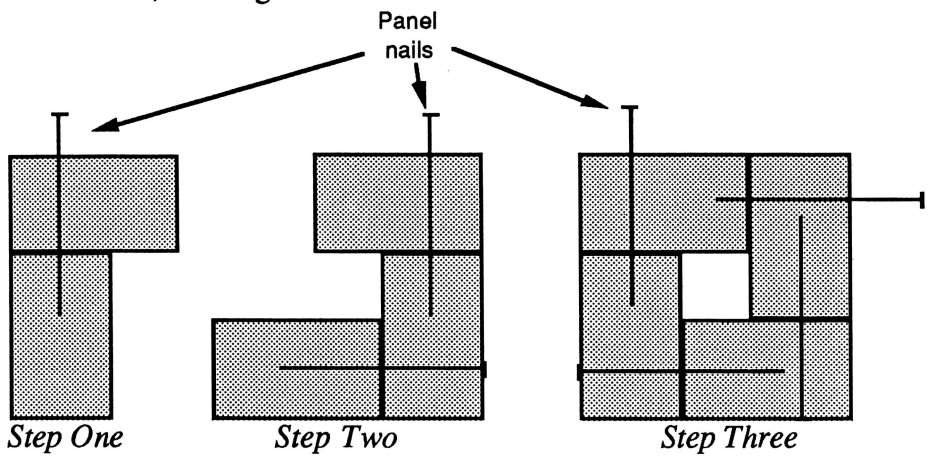




Note! Wear safety glasses and be extremely careful when making these cuts. When making crosscuts on short pieces they may be kicked back at you by the spinning blade. Always use a stick to get pieces away from the blade. NEVER GET CLOSE TO THE BLADE WITH YOUR HANDS.

☐ **3. Spray one side of the wood blocks with flat black paint.** Apply one coat of paint and let it dry. Then apply a second coat. The black side will be the inside of your colorimeter.

☐ **4. Assemble the block.** Before you begin, try holding the four blocks together as shown in the diagram below. Notice that the completed block will have a square hole in it which should be exactly the right size for the cuvette. Make sure the cuvette will fit into the hole. The first step is to start two nails in one piece about 3/8 inch in from each end of one piece. Form an "L" with another piece of wood and nail it securely. Start two nails in another piece and nail it to the "L" piece, forming a "U"-shaped piece. Check again to make sure the cuvette fits inside the "U". Start two nails in the last piece and nail it to the "U", forming a box.

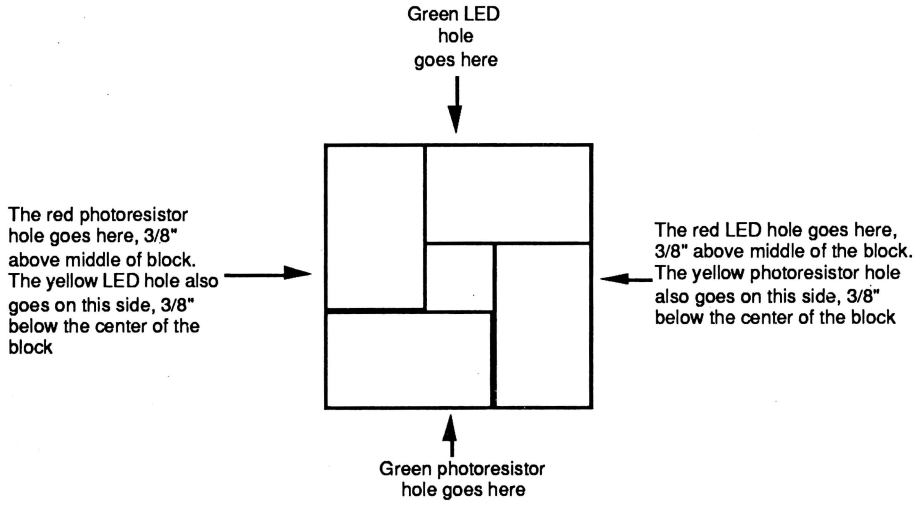


Colorimeter Being Assembled, Top View

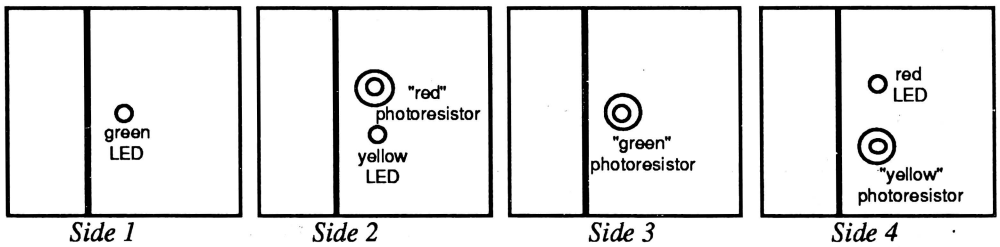
☐ **5. Drill holes in the block for the LEDs and photoresistors.** In this step you will be drilling three holes which extend through both sides of the block. An LED will be placed in one side of each hole. The light from the LED will shine through the hole to a photoresistor mounted at the other end of the hole. The green LED and photoresistor will use two opposite sides of the block. The red and yellow LEDs and photoresistors will share the other two opposite sides. The holes for the red LED and "yellow" photoresistor go in one side. The holes for the yellow LEDs and "red" photoresistors go in the other. All of these holes are centered vertically on the block. The red holes should be 3/8 inch above the center of the block and the yellow holes 3/8 inch below the center (see the two sets of diagrams below).

Drill the holes for the photoresistors first. These three holes should be the diameter of the photoresistors and should be only 1/4 inch deep. Be careful not to drill too far into the piece of wood. You will be drilling 3 holes in 3 different sides of the block.

Next drill the holes for the three LEDs. These holes should be exactly the diameter of the LED and smaller than the collar at the base of the LED. Drill these through both sides of the block starting in the center of the photoresistor holes. Keep the drill bit perpendicular to the block.

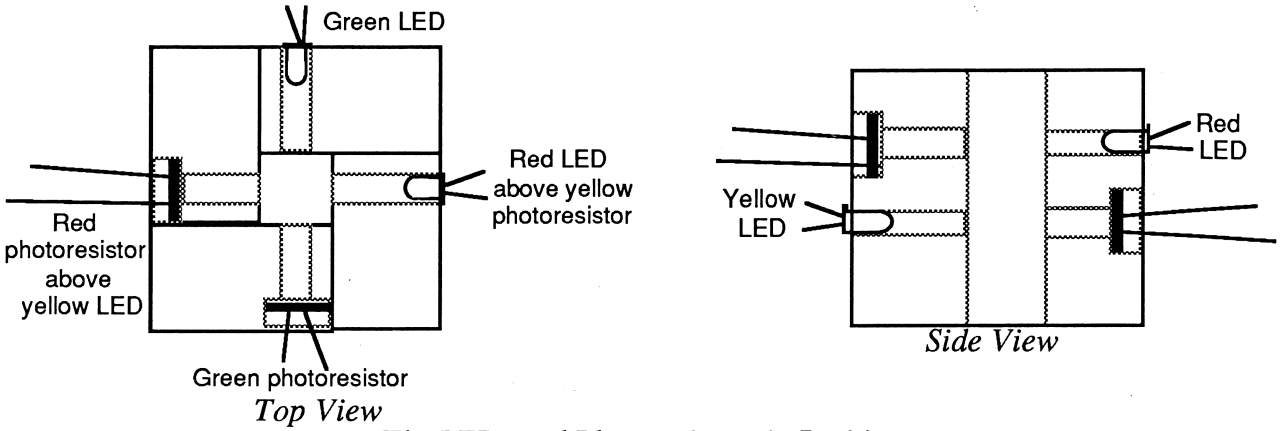


Top View of the Colorimeter, Showing the Holes for the LEDs and Photoresistors



Side Views of the Four Sides of the Block

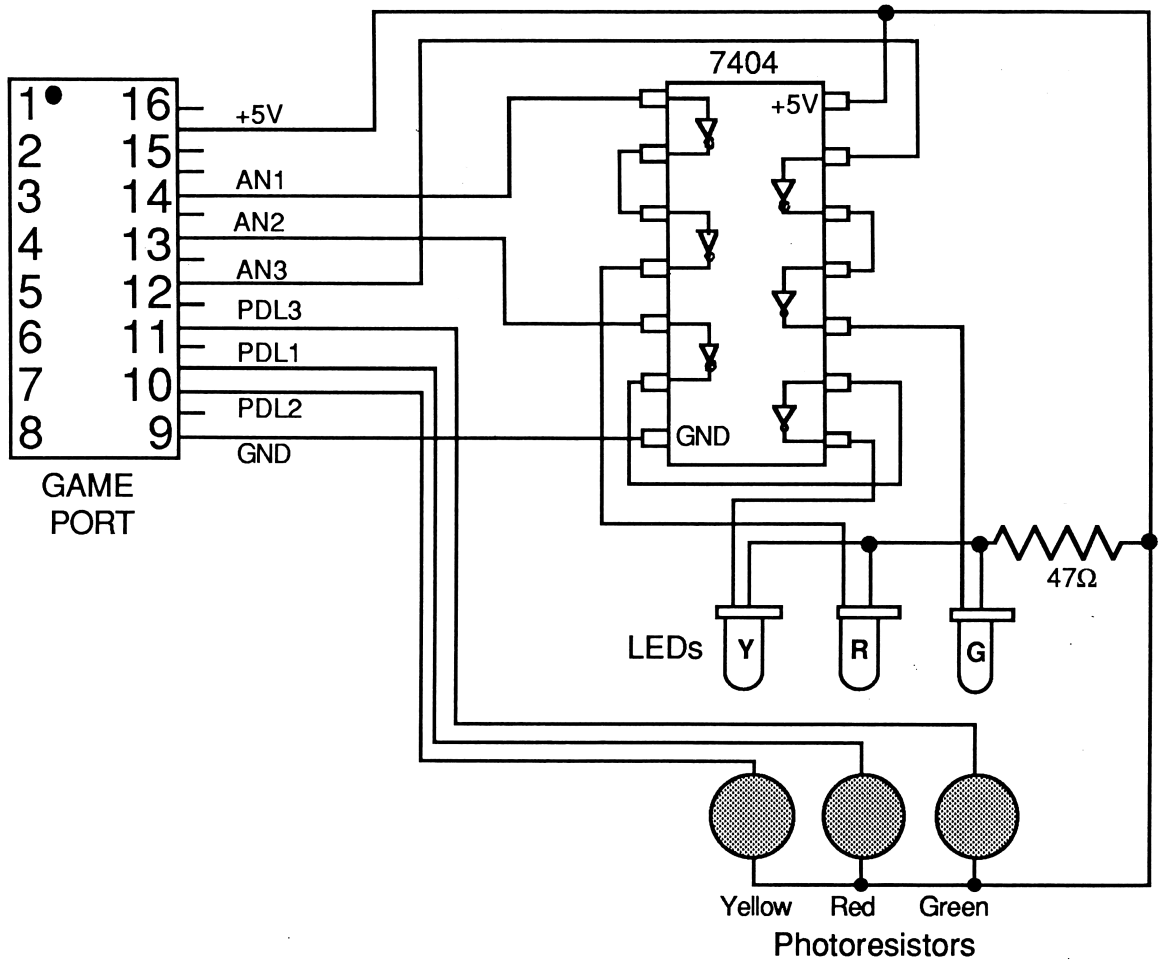
☐ **6. Glue the LEDs and photoresistors into the block.** Mix up a batch of epoxy putty the size of a grape. Mix it well and then divide it into 6 equal portions (about pea sized). Glue the photoresistors and the LEDs into place. Push each photoresistor into a hole and use a pencil eraser to tamp the epoxy putty in over the photoresistor. Place the green LED in the side with one hole and place the red LED above the yellow LED on the other sides. Make a little mound of epoxy putty over each LED and scratch a "G", "R", or "Y" in the putty to indicate which LED is in the mound. You will also need to know which side of the LED is the cathode. The cathode side has a flattened side on the collar. The other side is called the anode. Put a dot (pencil point) in the putty near the lead going to the cathode (flattened side) of the LED. Let the epoxy harden overnight.



The LEDs and Photoresistors in Position

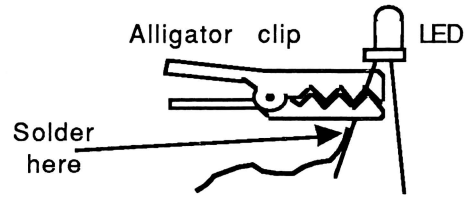
Part Two: Electrical Connections

A complete schematic diagram of the colorimeter is shown below. The LEDs and photoresistors are in the colorimeter block. The 7404 inverter IC and the resistor will be positioned on a small circuit board which will later be attached to the side of the colorimeter block. The 7404 IC is used as a buffer to provide enough current to light the LEDs without drawing too much current from the annunciator output lines. The resistor limits the current that flows through the LEDs. The connections to the 16-pin Apple II game port are made at the circuit board. The step-by-step instructions that follow will show you how to make all of these connections.



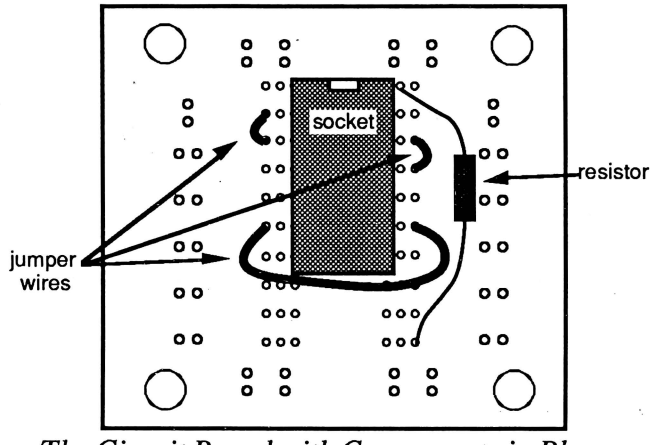
The Complete Schematic Diagram of the Colorimeter Project

☐ **7. Connect the LEDs together.** For this step and when connecting the cable to the LEDs, you should heat sink the leads of the LEDs to avoid damaging them. This is done by placing an alligator clip between the body of the LED and where you are soldering as shown in the picture below. Use two pieces of wire to connect the anodes of all three LEDs. The anode lead is the lead away from the flattened side of the LED and away from the dot marked in step 6. Solder the wires to the LED leads as close to the body of the LED as possible (allowing space for the alligator clip). Cut off any excess LED lead that extends past your solder connection. For neatness, keep these wires as short as possible.



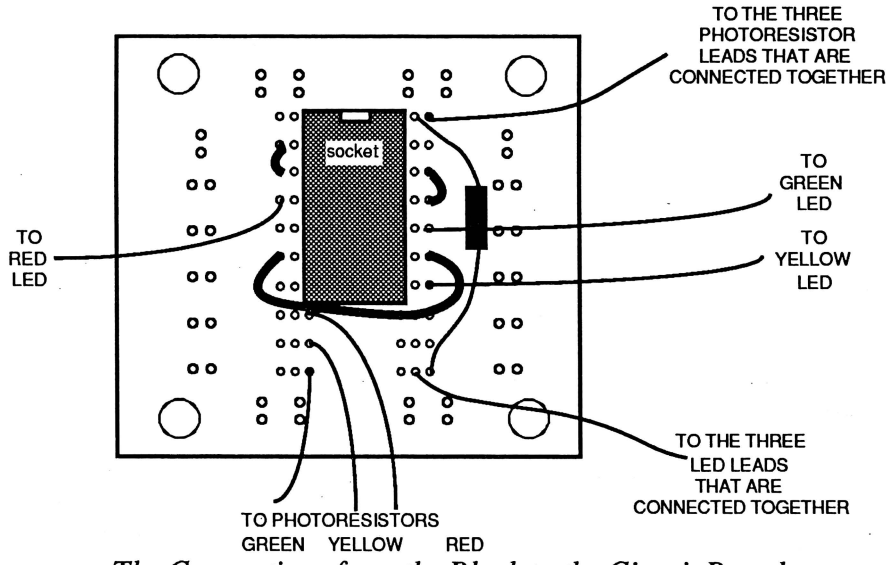
☐ **8. Connect the photoresistors.** Use two wires to connect one lead from each of the three photoresistors the same way you connected the LEDs together. There is no "+" side on a photoresistor, so you do not need to worry about which lead to use. Solder the wires in place and cut off any excess photoresistor lead.

☐ **9. Prepare the circuit board.** The diagram below shows how to position the 14-pin socket, three jumper wires, and the 47-Ω resistor on the circuit board. The circuit board shown is one-half of a Radio Shack #276-159. Other circuit boards can, of course, be used. Note that the notch of the socket (marking where pin #1 of the IC will be placed) is at the top of the diagram. Position the socket and other components on the plain side of the circuit board and solder them to the plated side of the board.



The Circuit Board with Components in Place

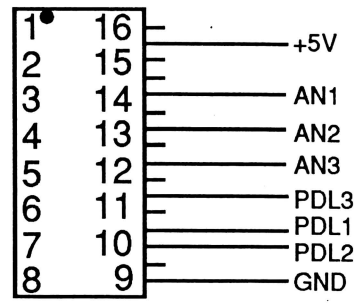
☐ **10. Make connections from the circuit board to the block.** Eight connections are needed between the circuit board and the colorimeter block. When the project is completed, the circuit board will be attached to the side of the colorimeter block, so these wires only need to be about 6 inches (15 cm) long. One of the wires goes to the common lead of all three LEDs. Another goes to the common lead of all three photoresistors. The other six go to the remaining leads of the LEDs and photoresistors. All of the connections on the circuit board are shown on the diagram below. Make the connections as neat as possible, and make sure that the leads from the LEDs and photoresistors do not touch together.



The Connections from the Block to the Circuit Board

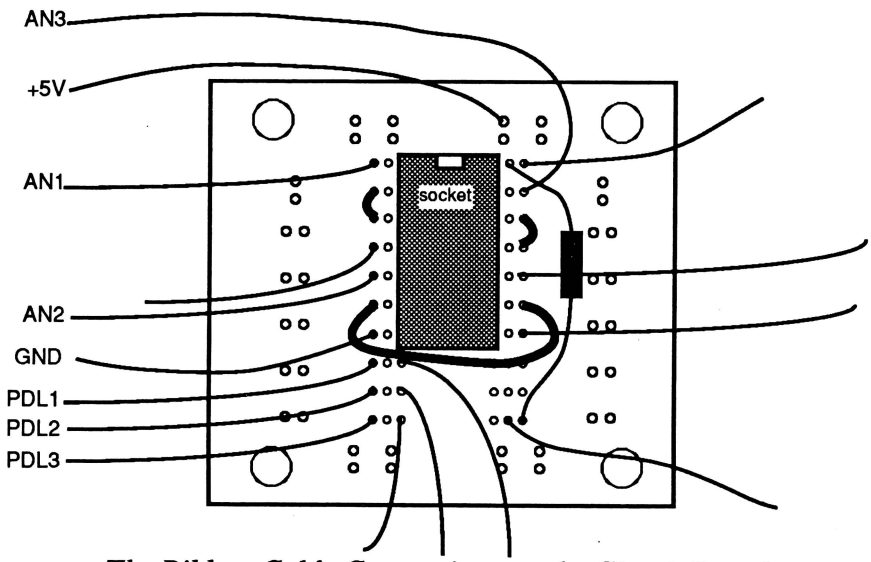
☐ **11. Prepare the 16-connector ribbon cable.** The following eight lines will be used: +5V, GND, AN1, AN2, AN3, PDL1, PDL2 and PDL3. The red LED is connected to AN1 and the PDL1 photoresistor is opposite the red LED. AN2 and PDL2 are used with the yellow LED and photoresistor. Finally, AN3 and PDL3 are used with the green LED and photoresistor. Prepare the 16-connector cable using the diagram below as a guide. Cut off the unused wires to get them out of the way. Strip about 1/2 cm on each of the wires you are going to use, being careful to avoid cutting any of the fine wires inside.

Pin	Connection
1	+5V
8	GND
10	PDL1
7	PDL2
11	PDL3
14	AN1
13	AN2
12	AN3



The Leads on the Ribbon Cable

☐ **12. Connect the ribbon cable to the circuit board.** Use the following diagram to show you where to connect the eight leads of the ribbon cable.



The Ribbon Cable Connections on the Circuit Board

☐ **13. Plug in the IC.** Carefully place the integrated circuit into the socket. Be sure to orient pin #1 at the proper corner of the socket (to the left of the notch). Make sure all 14 pins go into the socket.

☐ **14. Check the block with the test program.** Connect the ribbon cable to the Apple game port, making sure that pin #1 is oriented properly. Place the *Chaos in the Laboratory Programs #1* disk in the disk drive and turn on the computer. If the computer does not start up normally, turn it off and go to the "Troubleshooting" section below. RUN the program COLOR.TESTER with no cuvette in place. This is a simple test program that will turn on each of the LEDs in order and report the reading from the photoresistors. Cover the top of the block to keep out room light. When the LED opposite a photoresistor is on, the count reported should be fairly low (usually less than 100). When the LED is off, the count should be 255. Step through the program, examining the counts displayed for each LED. Make sure that the appropriate photoresistor is responding for each LED. If you get a count of 0, you probably have a short between two of the leads of the photoresistor. If you get a count of 255 when the LED is on, you either have the wrong LED opposite the photoresistor or you have a break in the photoresistor circuit. If the counts are higher than you expect all of the time, perhaps you have light leaking into the block through the LED and photoresistor holes. Try turning off the room light. That should not make any difference. If it does, block out the light by using black electrical tape or opaque glue.

You can also visually check to make sure that the LEDs are going on at the right times. Turning off the room lights and putting a thin strip of paper down the cuvette hole makes this easier.

☐ **15. Make a calibration curve.** Add a few drops of red food coloring to a glass (250 ml) of water. Now run the program COLOR, which is also on the *Chaos in the Laboratory Programs #1* disk. You will be asked to insert a cuvette containing just water. This is so the program can make a measurement of the light absorbed by the cuvette and the water alone. Next you will need to select the color of the LED to be used. Since you are using a red solution, which transmits red light, the green LED would probably be the best to use. Place your red solution in the cuvette and press <RETURN> to take a reading. If the third column reading is 255, your "standard" is too concentrated. If the reading is low (less than 80), add additional drops of food coloring. Experiment with the solution until you get third column readings between 80 and 250. After you have established a good starting point, make a series of dilutions from 10 to 90 per cent as concentrated as the "standard." Use a graduated cylinder (or pipets and volumetric flasks). Measure the absorbance of each solution. Make a graph (on paper or using a computer graphing program) of the results with concentration (percentage of the standard) on the x-axis and absorbance on the y-axis. You should get a straight line with a near-zero intercept.

After you have done your calibration graph, you should be able to measure the concentration of an "unknown" solution. Have someone else mix a dilute solution from your standard. They should keep track of the amount of water and standard solution used. You should be able to determine the concentration by using the colorimeter to measure the light absorbed by the solution.

The same solution can also be studied using the yellow LED. A red solution will transmit more yellow light, so the variation in readings will be less. You may also want to experiment with the AUTOMATIC feature of the COLOR program. If you select AUTOMATIC for the color of the LED to be used, the program will take measurements and select the best LED to use for the solution you are using.[5]

☐ **16. Complete the construction of the block.** If the colorimeter seems to be working correctly, use two small tacks to attach the circuit board to the side of the colorimeter block. If you prefer, you can use double-sided tape, or even a rubber band for this job. Also, you should tape over the wires to make the apparatus less fragile. To keep the block from turning over as it is used, you can

---

[5]What the AUTOMATIC feature does is test the light transmitted from each of the three LEDs, one at a time, and select the one with the least transmitted light.

attach a base (about twice the size of the block) to the bottom of the block. It should be painted flat black. A small block to cover the hole (to keep out room light) that has been painted flat black is also a good idea.

Troubleshooting

◊ If the LEDs do not work properly, check the following:
- Make sure the leads of the LEDs are being used properly
- Make sure the 7404 IC is wired properly.
- Make sure that the three common leads of the LEDs connect to the resistor and the other end of the resistor connects to the +5V line.

◊ If the LEDs are all working but not corresponding to the color specified in the program, you have two options. You can redo the connections to the computer or change the basic program so that the LEDs are turned on correctly. If one or more of the LEDs are not working, check their connections to the computer. Make sure the correct side of each LED is connected to the +5V line from the computer.

◊ If the LEDs work but the colorimeter program does not, it could be that the correct LED is not opposite the designated photoresistor. You could have also connected the photoresistor incorrectly. Measure the resistance of each photoresistor using the highest resistance setting on your ohmmeter. You can measure this at the 16-pin connector between pin #1 and each of the PDL pins (pins #10, #7, and #11), or directly on the colorimeter block. Measure the resistance of each photoresistor both with and without a bright light shining in the cuvette hole. The resistance should change when you turn the light off. If you get a bad reading, check the wires and connections. Replace a bad photoresistor if necessary.

◊ If all the LEDs and photoresistors test out but the system still does not work properly, there are three things that could be wrong: (1) The wrong LED is opposite the designated photoresistor. RUN the program COLOR.TESTER and it should tell you if that is the case. If this is the problem, correct the wiring or change the program. (2) The photoresistor has too low a sensitivity for the amount of light coming from the LED. The program COLOR.USR expands the range of the light levels and should correct the problem of too little light. Refer to the "Supplemental Information" section for information on this program. (3) Finally, the Apple computer itself could have a bad game port. This would require servicing at your Apple dealer.

Project Extensions

Program Modifications

◊ The COLOR program can be modified to include a calibration section in which you enter known concentrations and the instrument takes absorbance data. After calibration, the program could then display concentrations as well as absorbance.

◊ Write a program for examining the absorbance of two different wavelengths of light (two different LEDs). The program should turn on one LED and make a measurement at that wavelength for a few seconds, and then it should turn that LED off and a different one on. This program can be used as a "double-checked" method of analysis for pure solutions, or it can even be used to analyze mixtures of two different solutions. Red and blue food coloring will illustrate the concept. After readings are taken separately for two pure solutions, the new program combined with some algebra can be used to find the concentrations of the mixture. Both the red and blue (pure solutions) are measured separately using both the green and the yellow LEDs. From the absorbances, two absorptivities (a) are calculated for each solution. The colorimeter is then used to measure the absorbance of a mixture. Since absorbance is additive, the equations below can be used to find each component's concentration. The subscripts indicate first the LED used and then the component ($\mathbf{a_{gr}}$

is the absorptivity when the green LED is used with the red solution). The two equations can be solved using the method of simultaneous equations to find the concentration of both the red and blue food coloring in a mixture.

$$A_y = a_{yr}\, l\, c_r + a_{yb}\, l\, c_b \quad \text{and} \quad A_g = a_{gr}\, l\, c_r + a_{gb}\, l\, c_b$$

Since the path length is the same for all solutions the equation can be reduced to:

$$A_y = a_{yr}\, c_r + a_{yb}\, c_b \quad \text{and} \quad A_g = a_{gr}\, c_r + a_{gb}\, c_b$$

Circuit Modifications

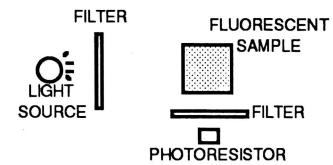
◊ Make a 4-channel colorimeter by including an infrared LED as the fourth "color." AN0 and PDL0 were not used in the original project and are available for the fourth channel. Some solutions absorb IR light and can therefore be determined by this type of colorimetry. The photoresistor used in the main project is sensitive to infrared light also, and infrared LEDs can be found at Radio Shack. You will need to modify the programs to use the fourth LED and photoresistor.

◊ Just recently, blue LEDs (peak wavelength 470 nm) have become available. Adding a blue LED to the circuit would be a nice addition. It could either take the place of the yellow LED or be added as a fourth color controlled by the AN0 line. Be forewarned that blue LEDs are much more expensive than other LEDs.

◊ A different type of colorimeter can be made using a white light bulb and colored filters to restrict the wavelength of light. The basic plan is similar to that used in the LED colorimeter. You should make a wooden block that has a hole down the middle for the light to shine through. The white light source should be at one end of the hole, with the photoresistor at the other. You will need to make a square hole to hold the cuvette and a slot to hold the filter. The light should pass through the filter before it reaches the cuvette. To analyze darker samples, a bright halogen light could be used. Different photoresistors have very different wavelength sensitivity. Most of the inexpensive ones are too responsive in the infrared region for analyzing most solutions. A 1 gram per 100 milliliter solution of copper sulfate ($CuSO_4 \cdot 5H_2O$) acts as a filter for infrared light. You could add a hole for a second cuvette or vial of copper sulfate to filter out most of the infrared light. This should be placed close to the light source before either the filter or the sample cuvette.

◊ More advanced commercial spectrophotometers measure the absorption of light over a range of wavelengths. A challenging, but interesting project begins with using a prism or diffraction grating to produce a spectrum. By moving the photoresistor, a range of wavelengths can be analyzed. You could use the stepper motor from Project 11 of this book to control the movement so that the computer can tell what wavelength is being scanned. The response of a photoresistor varies greatly with wavelength, so you will have to calibrate it to read correctly. The photoresistor used in the main project will work, but you may want to experiment with different light sensors. Many different photosensors are available. Each has a particular wavelength range over which it is most sensitive. Hamamatsu Electronics (phone 908-231-0960) is a good source for a wide variety of photoresistive cells and photodiodes. They will send you a catalog which contains technical and application information.

◊ A fluorometer is used to measure the light produced by a substance as it fluoresces when exposed to a light source. The wavelength of the light produced is not the same as the wavelength of the light source. To measure this, you need the light source at a right angle to the detector. A second filter slit would also be needed.



The lamp for fluorescence should be much brighter; therefore, the Apple II power supply should not be used to power it. A small, quartz halogen lamp with an independent power source should be used. When measuring fluorescence the wavelength of light that shines on the sample should be of a shorter wavelength than the light glowing from the solution. Usually a blue filter is used on the

lamp side and a longer wavelength filter is used on the photoresistor's side. This way only the light produced by the sample will reach the detector. Absorbance is not used in fluorescence; instead the light intensity itself is proportional to concentration. New software will be required. Remember, the PDL readings go down as the light intensity goes up, so a reciprocal must be taken to record light intensities. Many organic chemicals such as aspirin and quinine (in tonic water) will fluoresce.

Additional Experiments

CHLORINE ANALYSIS IN WATER SAMPLES

Colorimetric procedures can be used for many chemical determinations. A suitable one where almost everyone can get satisfactory results involves using reagent pillows for determining the chlorine levels in water samples (DPD reagent pillows for total chlorine, catalog #14064-99, Hach Chemical Co., P.O. Box 389, Loveland, CO 80539, (800) 227-4224) . The contents of one reagent pillow are mixed with 25 ml of water; the intensity of the color that develops is proportional to the chlorine in the water. A fresh bottle of Clorox bleach contains 5.25% sodium hypochlorite (5.25 g NaClO / 100 ml) or 2.50% chlorine.

$$5.25 \text{ g NaClO / 100 ml} \times \frac{1 \text{ mole NaClO}}{74.5 \text{ g NaClO}} \times \frac{1 \text{ mole } Cl_2}{2 \text{ mole NaClO}} \times \frac{71.0 \text{ g } Cl_2}{1 \text{ mole } Cl_2} = 2.50 \text{ g } Cl_2 \text{ / 100 ml}$$

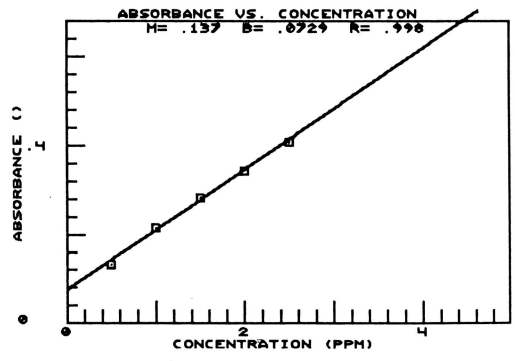
2.50% is 2.50 g / 100 g solution or 2.50 g / 100 ml when water is the solvent. Chlorine concentrations are usually reported in "parts per million," which is grams of material per million ml (or g) of water. 2.50% is 25,000 parts per million (ppm). Normal treated water has 2.5 ppm chlorine or less. To make standards for your calibration curve, dilute the fresh Clorox 5.00 ml to 500 ml. Mix well and dilute the mixture a second time 5.00 ml to 500 ml. This produces a solution with a concentration calculated below:

$$25{,}000 \text{ ppm} \times \frac{5 \text{ ml}}{500 \text{ ml}} \times \frac{5 \text{ ml}}{500 \text{ ml}} = 2.5 \text{ ppm}$$

> Caution: If chlorine bleach gets in your eyes it can cause damage. Use caution and wear safety goggles when pouring and mixing chlorine bleach.

Now make up some dilutions of your standard solution for calibration. Add one DPD reagent pillow to each of the following solutions:

2.5 ppm = 25 ml of the second dilution.
2.0 ppm = 20 ml diluted Clorox + 5 ml water
1.5 ppm = 15 ml diluted Clorox + 10 ml water
1.0 ppm = 10 ml diluted Clorox + 15 ml water
0.5 ppm = 5 ml diluted Clorox + 20 ml water
blank = 25 ml distilled water



A sample calibration graph using these standard solutions is shown above. The blank used for this series of measurements was the distilled water with the DPD reagent pillow added.

Treated city water, swimming pools, and most suburban water systems contain measurable levels of chlorine. The chlorine level varies with temperature, time since the last treatment, and distance from the treatment facility. Studies involving water from various homes and even water from a common source could be made. Since chlorine is put in the water to kill "germs," there could also be applications in biology projects.

OTHER EXPERIMENTS USING THE COLORIMETER

Many types of chemical analysis can be carried out with your colorimeter. Check in books on analytical chemistry and quantitative analysis. The Hach catalog is a source of convenient reagents for doing a variety of analyses related to the environment. Several examples are given below:

- Manganese in stainless steel is determined by dissolving the steel in nitric acid and reacting the solution with potassium metaperiodate (KIO_4). The manganese is oxidized by the periodate to permanganate, which is an intensely purple-colored solution. The absorbance of the solution is proportional to the amount of manganese.
- Copper in water can be determined by reacting the solution with a special organic reagent (diethyldithiocarbamate). A metal complex is formed that is yellow, so the green LED should give reasonable results.
- Iron at 5 part per million or higher can be determined by reacting the iron with hydroxylamine hydrochloride followed by 1, 10 phenanthroline. The color developed is orange.

Caution: Read labels and follow safety precautions when using chemicals.

CHEMICAL KINETICS

Another area of chemistry that can be investigated with your apparatus involves *chemical kinetics* (the speed of chemical reactions). The formation of chemicals or the rate chemicals are used during chemical reactions is an important area in chemistry.

Mix 1 drop of red food coloring with 25 ml water. Add a few drops of Clorox and mix the solution rapidly. Pour a portion into your cuvette. Place your cuvette in the block colorimeter and measure the absorbance every 15 seconds until the readings level off. As the chlorine in the bleach reacts with the red dye, the solution becomes more transparent. Try the experiment with varying amounts of Clorox and compare results. Temperature has a great effect on the rate of chemical reactions, so you could repeat the above experiment at various temperatures.

A more impressive approach would be to write a program with delay loops so you can plot solution absorbance versus time on the x-axis. Be sure to include a section for setting the blank (P_O). RUN your program to set the blank and then prepare your kinetics chemicals. Have a GET statement delay until you have the chemicals in place before you start the plot.

Here are some other examples of how the colorimeter can be used to follow the progress of chemical reactions that involve a color change:

- Potassium permanganate reacts slowly with oxalic acid in the presence of sulfuric acid. A mixture of oxalic acid, sulfuric acid, and potassium permanganate is initially deep purple but the color fades in time to nearly colorless. Measuring color intensity and time is a method to investigate the rate of reaction. Changing the temperature or adding a catalyst will alter the rate of reaction.
- Formaldehyde reacts slowly with sodium bisulfite and the reaction mixture becomes more basic with time. In the presence of an acid-base indicator, we end up with a solution that changes color.
- The pigments in deciduous trees decompose in a few days. This decomposition can be studied by measuring the decrease in color intensity over a week or so.

Supplemental Information

Program Design Notes

The COLOR.USR Program

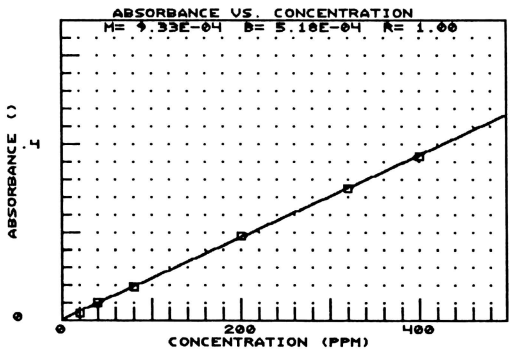
The program COLOR.USR is included on the *Chaos in the Laboratory Programs #1* disk. It expands the range of the light sensitivity for the colorimeter. If you use very concentrated solutions, this program may be useful since it can read lower light intensities. The disadvantage is that this program

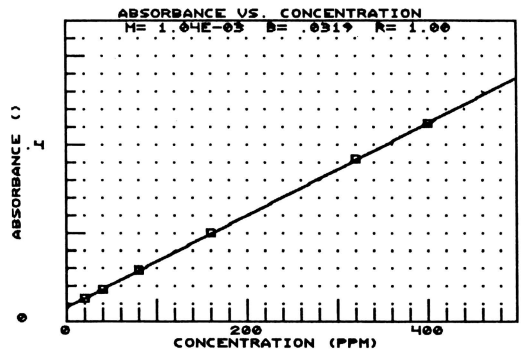
takes longer for absorbance readings. The COLOR.USR program uses a seldom used command USR, which calls a user-defined routine. The machine-language routine XP sets up the user-defined routine, and must be on the same disk. In using the program you select a range for each PDL input by entering a number between 2 and 32. The normal range for a PDL count is 0 to 256. If you enter 2, that range will be doubled, but each reading will take twice as long. You will have to experiment to determine the range necessary for your experiments. Remember the higher the number selected, the slower the program will run.

Limitations of the Colorimeter

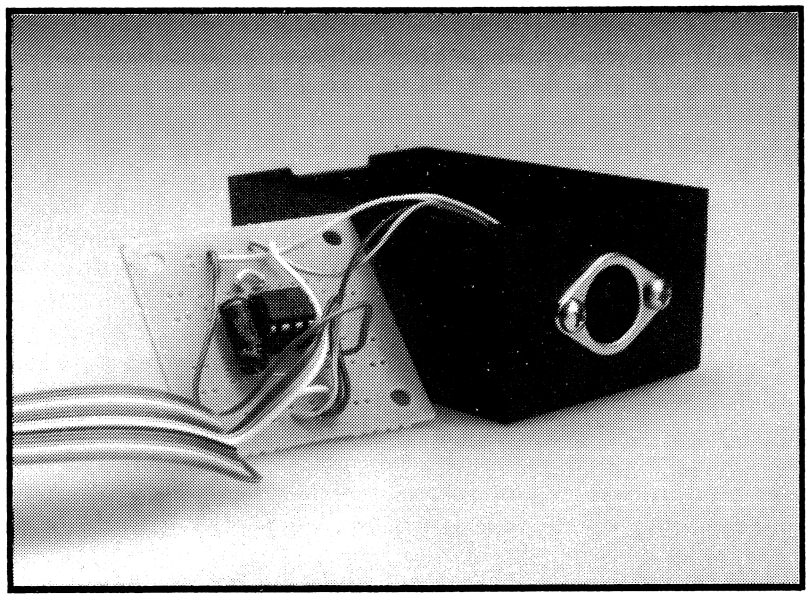
The COLOR program makes a large assumption in its calculation of absorbance. It assumes that the "count" reported by the PDL command is inversely proportional to the light intensity. This is an approximation. The count reported by the PDL command is directly related to the resistance of the photoresistor. The resistance of most photoresistors decreases as light intensity increases. Unfortunately the relationship between resistance and light intensity is not a simple one. Over a narrow range of light intensities, it is approximately an inverse relationship. The best way to verify that the assumptions made here are valid for your particular photoresistors is to do a Beer's law experiment. This is why you are encouraged to do this experiment as soon as you have your colorimeter working. If the Beer's law graph produces a straight line with a zero y-intercept, the assumption made about the response of the photoresistor is justified.

Here are some sample graphs we have made using this colorimeter. The left graph shows a Beer's law experiment using blue food coloring and the yellow LED. The right graph was made using green food coloring and the red LED. As you can see from these graphs, the assumptions mentioned above give a reasonably good approximation of the results observed.





Sample Beer's Law Graphs



Project 5

A-to-D Converter

by J. Roger Bacon

This project consists of building an inexpensive analog-to-digital converter (ADC) capable of measuring voltages very quickly. This device connects to the 16-pin Apple II+, IIe or IIGS game port. The heart of this circuit is a 10-bit Serial Analog-to-Digital Converter IC. This IC reads the voltage, converts it to a digitally-coded number, and then sends the number to the computer using serial data transfer.

This project (or an alternative voltage input device, such as a Voltage Monitor) is a prerequisite for several of the other projects in this book: Barometer, Magnetic Field Sensor, Light Meter, and Smart Battery Charger. This project is also required if you want to digitize sounds for use with the D-to-A Converter/Digitized Sound project.

The *Chaos in the Laboratory #2* disk includes programs for operating the A-to-D Converter. One program simply displays the voltage on the screen, and the other turns the Apple II into an oscilloscope, displaying 6000 voltage readings per second. This program can be used to study the patterns produced by various sound waves or for many other experiments traditionally requiring an oscilloscope.

Project Requirements

Computer

This project uses the annunciator output lines on the 16-pin Apple II game port. These lines are available on the II+, IIe, and IIGS. This project cannot be done on other models of Apple II computers.

Background Knowledge

- Electronics: Experience with soldering components on a circuit board is recommended.
- Working with tools: If you are building this project using a Vernier Software parts kit, a predrilled project box is included. Otherwise, experience with drills and hand tools is recommended for modifying the project box.
- Programming: No programming is required for the core project. If you want to modify the programs, you can probably do so by changing just the Applesoft BASIC portions of the programs. Making changes to the two machine-language routines used to actually read the A-to-D Converter would be more difficult and require knowledge of assembly language.

Parts

- ☐ LTC1092 ADC chip, manufactured by Linear Technology and distributed by Marshall Industries, 5224 Greens Darry Road, Raleigh, NC 27604, (919) 878-9882. Marshall's telemarketing group number is 1-800-522-0084. Marshall has a $25 minimum order requirement. Newark Electronics, Arrow Electronics, and other electronics supply houses may also be able to provide the LTC1092 chip. Their addresses and phone numbers are listed in the "Sources of Electronic Components" section of the *Introduction.*
- ☐ 8-pin DIP socket (Radio Shack #276-1995)
- ☐ circuit board (one half of a Radio Shack #276-159)
- ☐ 10-µF, electrolytic capacitor rated at 6 volts or more (Radio Shack #272-1013)
- ☐ 16-conductor ribbon cable with a DIP connector on one end
- ☐ small project box (Radio Shack #270-230)
- ☐ 5-pin DIN socket (Radio Shack #274-005)
- ☐ 5-pin DIN cable (Radio Shack #274-003 with three wires connected or half of a #42-2151 patch cable)
- ☐ (2) test clips (Radio Shack #270-372)
- ☐ (2) short jumper wires for connections on the circuit board
- ☐ (3) wires 10-cm long for connecting the DIN socket to the circuit board

The Vernier Software A-to-D Converter Parts Kit includes all of these parts. It also includes a plastic box and lid for enclosing the circuit. This kit sells for $32.00 (Order Code CAD-16).

Tools and Supplies

Soldering iron, solder, multimeter (for measuring DC voltage and resistance), screwdriver, electric drill and drill bits, and a flat file. The electric drill and flat file are not needed if you are using a Vernier Software parts kit.

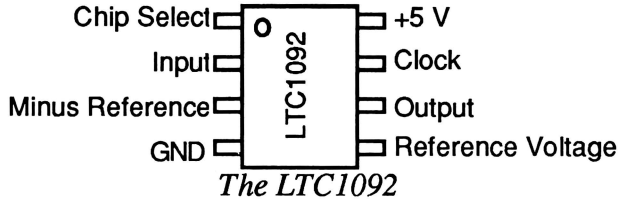
Background Information

This project consists of building an analog-to-digital converter (ADC) capable of taking voltage readings very quickly. Most commercially available laboratory interfacing units contain ADCs. These devices measure analog voltages (continuously variable values) and convert the voltages into digital numbers the computer can understand and work with. There are many types of ADC integrated circuit chips available. They vary in terms of the mode of operation, the accuracy of the digitized signal, speed, output, and cost.

All ADCs report the voltage measured as a *count*. A key feature of an ADC is the number of binary bits included in this count. Eight bits is usually the minimum number used. An 8-bit A-to-D converter reports a voltage as a count that can vary over a range of 0 to 255 (there are 256 possible values because $2^8 = 256$). A 10-bit ADC is used in this project. It can report a voltage as any of 2^{10} possible values, or in the range 0 to 1023 counts. For this project, the highest voltage that can be read is +5V. So with a 10-bit ADC, the voltage is read in steps of (5 volts/1024), or approximately 5 mV. Twelve, 14, 16 bit, and recently even higher precision ADCs are available. The cost of ADCs goes up as the number of bits increases.

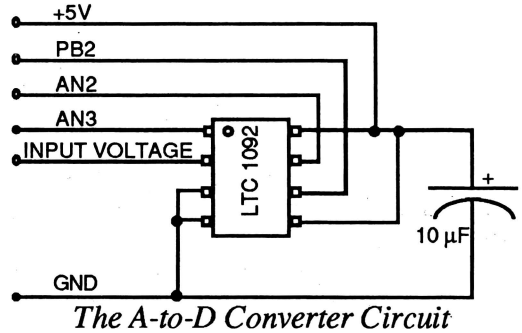
This interfacing project is based on the LTC1092 10-bit Serial ADC manufactured by Linear Technology, Inc. It is inexpensive, yet it still provides reasonably fast voltage measurement. The A-to-D Converter we are going to use can take data as fast as once every 20 microseconds (µs), but the speed of the Apple II computer and the machine-language routines used to control it in this project limit the speed to about one sample every 160 µs. This is still fast enough to take about 6000 voltage measurements per second.

The ADC integrated circuit is connected to the computer via the 16-pin game port.[1] The LTC1092 chip can be powered, controlled, and monitored using five lines from the Apple computer. Since the output is serial, the data flows from the chip to the computer on one wire instead of the eight or more used in the more common parallel output ADCs. One of the game port push-button (PB) inputs is used for this purpose. The LTC1092 also requires two control lines from the computer. One annunciator output line from the game port serves as the clock signal and simply turns on and off at a software-controlled rate. Another annunciator output tells the chip when to take and send data (chip select). The two remaining connections are for +5V and ground power connections. The chip requires only 2.5 mA of current to operate, so there will not be a significant load on the Apple II power supply. The pinout of the LTC1092 is shown below:



The LTC1092

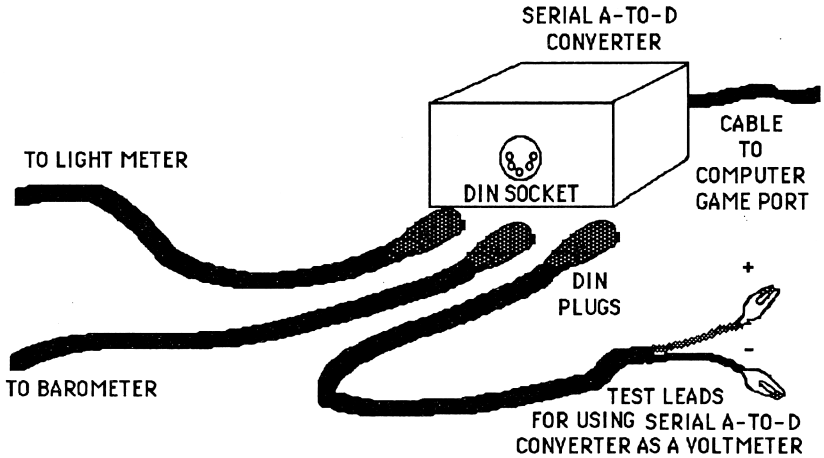
The complete schematic of the A-to-D Converter is shown here:



The A-to-D Converter Circuit

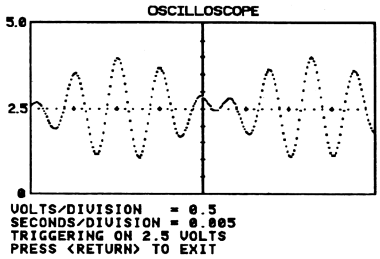
[1]The output lines needed are not available on the 9-pin external game port. This is the reason that this project cannot be done using an Apple IIc, IIc+, or Laser 128 computer.

We suggest placing the circuit in an experimenter box and mounting a 5-conductor DIN socket in the side of the box. The A-to-D Converter can then be used in several different ways. If you connect the simple 2-conductor cable with test leads on the ends, it can be used for direct voltage measurements (like a voltmeter or an oscilloscope). Alternatively, you can connect other electronic circuits and study their voltage output. Projects 6, 7, and 8 are designed to be used in this way. They are circuits that connect to the A-to-D Converter in order to indirectly measure light level, air pressure, and magnetic fields. The same type of DIN socket is also used on the Vernier Software Voltage Input Unit and the Voltage Monitor from *How to Build A Better Mousetrap*. Devices built with the 5-pin DIN plug can be used with any of these different voltage measurement devices.



The A-to-D Converter Used With Other Circuits

Three programs for reading the voltage with the A-to-D Converter are provided on the *Chaos In the Laboratory Programs #2 disk*. The program A.TO.D is a BASIC program that simply displays the voltage. A short machine-language routine does the actual work of decoding the data from the IC into two 8-bit bytes that the computer can manipulate. Another BASIC program, A.TO.D.AVG also displays voltage, but it takes 64 quick readings and then displays the average voltage. The BASIC portion of these programs is fairly simple and can be modified as a project extension. The program SCOPE uses a complex machine-language program to display a voltage vs. time graph similar to the display of an oscilloscope. Since this program is written mostly in machine language, it can take data and display the results very quickly. The SCOPE program even allows you to set a "triggering level," as on a traditional oscilloscope. This program can show the waveform of a sound wave, as long as the frequency is not high (the lower notes on a keyboard). A sample screen showing the SCOPE program in action is displayed below:



A Sample Screen Using the SCOPE Program

There are a number of interesting extensions for this project. Two of them take advantage of other features of the LTC1092 Serial A-to-D Converter. The IC is externally referenced. This means the magnitude of the maximum signal that will be digitized is set by connecting that voltage to one of the pins as the reference voltage. The "count" read by the 10-bit device always corresponds to 1/1024 ($1/2^{10}$) of the reference voltage. For the core project, V_{ref} is 5 volts, so each count represents about 5 mV, but if the V_{ref} is only 500 mV, then the count resolution is 0.5 mV. This frequently eliminates the need for preamplification of experimental signals. A good example would be a thermocouple circuit. These devices give out voltages that are proportional to temperature. The voltages are typically in the 0

to 100 mV range. Setting V_{ref} at 100 mV would eliminate the need for the amplifier circuit usually used to step up this small voltage.

The LTC1092 also has differential inputs. That means that there are two input pins and the voltage reported is the difference between these pins. In the core project we suggest connecting the negative input to ground so that the positive input signal is measured with respect to ground. One of the project extensions is to set up the LTC1092 so that it measures a differential or "floating" voltage.

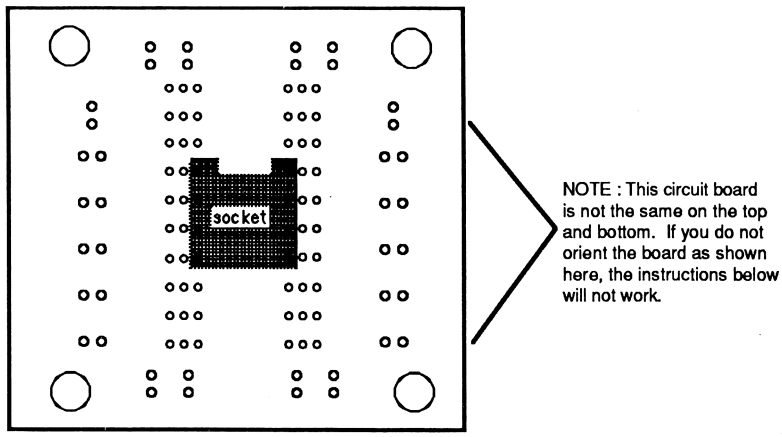
If you are interested in digitized sound experiments, this project and the D-to-A Converter (Project 10 of this book) can be combined to make a circuit that can digitally "record" a sound and then play it back. One of the project extensions for Project 10 explains how to do this. For other ideas, including program enhancements and suggestions for other experiments that can be done using the A-to-D Converter, see the "Project Extensions" section.

> The voltage range of this project is 0 to +5 volts DC. Connecting a voltage to the project outside this range could damage the LTC1092.

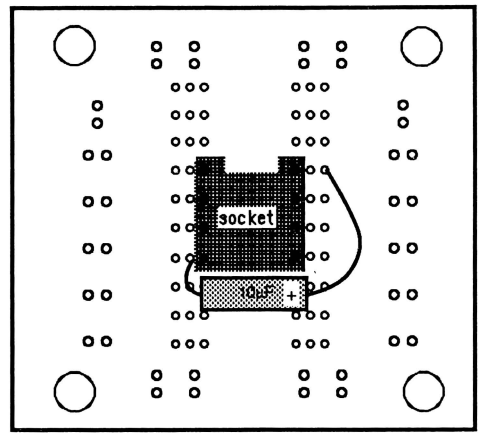
Step-by-Step Instructions

These instructions assume that you are using one half of a Radio Shack #276-159 circuit board. If you use a different circuit board, you will have to decide on the layout of the components yourself.

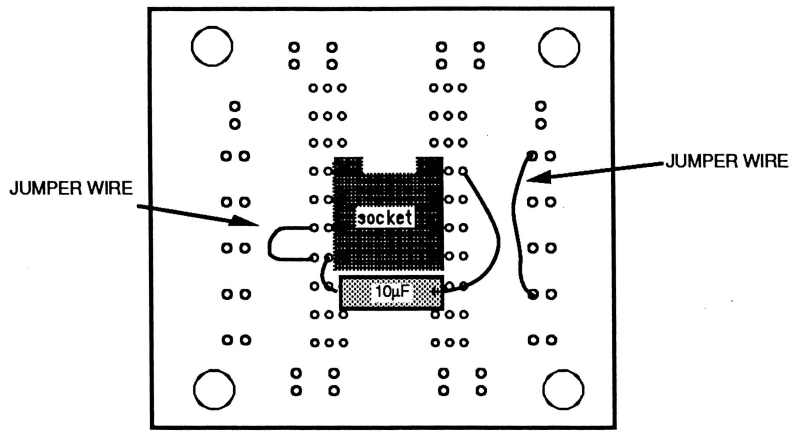
☐ 1. **Insert the socket.** Insert the 8-pin IC socket into the circuit board. The socket goes on the plain (no printing or copper plating) side of the circuit board. Note the notch on the socket which indicates pin #1. The notch should be toward the top of the circuit board. The pins of the socket are numbered in a counterclockwise pattern around the socket. Bend the pins enough to hold the socket on the board. Solder all 8 pins to the copper-plated side of the circuit board. Do not insert the IC in the socket at this time.



☐ 2. **Insert the capacitor between +5V and ground.** Be careful to observe the orientation of the capacitor. Solder the "+" lead to pin #8 and the "–" lead to pin #4 as shown on the diagram below. On most capacitors, the negative terminal is labeled "–".

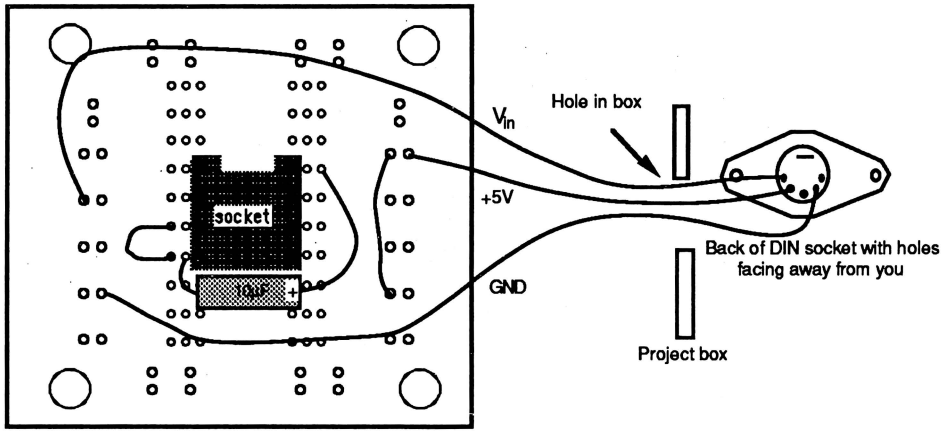


☐ 3. **Solder two jumper wires to the circuit board.** Solder a short jumper wire between pins #3 and #4 and another jumper wire between pins #5 and #8 of the IC as indicated below. Cut the jumper wires short to keep the circuit neat.



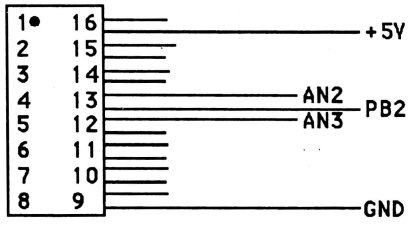
☐ 4. **Prepare your project box.** The 5-pin DIN socket will eventually be mounted in one end of your project box. You will need to drill suitable holes: a 5/8 inch diameter hole for the socket and two 1/8 inch holes for the mounting tabs. The box will also need a path for the ribbon cable at the end opposite the DIN socket hole. A flat file can be used to file a notch for the cable. If you are using a Vernier Software parts kit this step is already done for you.

☐ 5. **Connect leads to the DIN socket.** Solder three 10-cm long wires to the DIN socket. Run these wires through the hole in the box and connect them as indicated in the diagram below. Solder the wires in place.



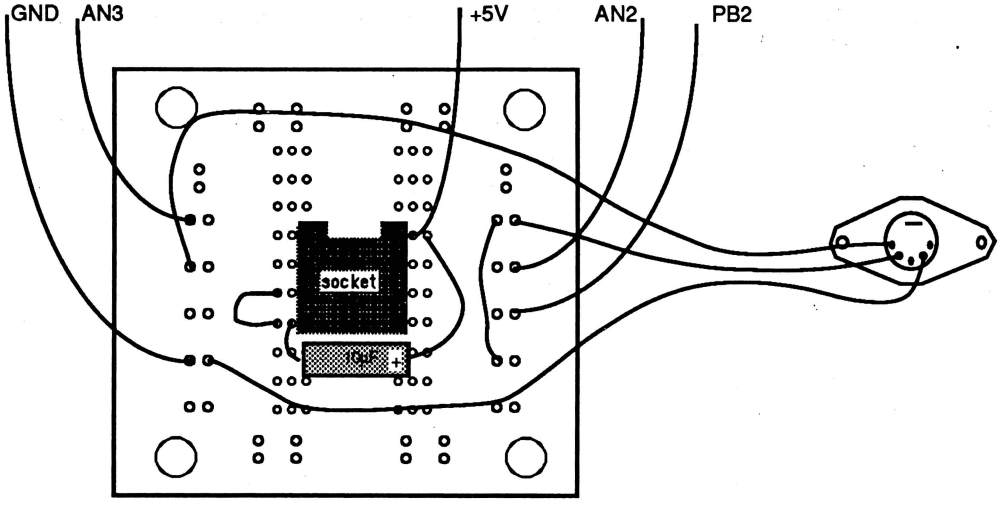
☐ 6. **Prepare the 16-conductor ribbon cable that will connect the circuit board to the computer.** Connections for AN2, AN3, PB2, +5V and GND are shown in the figure below:

Pin 1 = +5V
Pin 8 = GND
Pin 13 = AN2
Pin 12 = AN3
Pin 4 = PB2

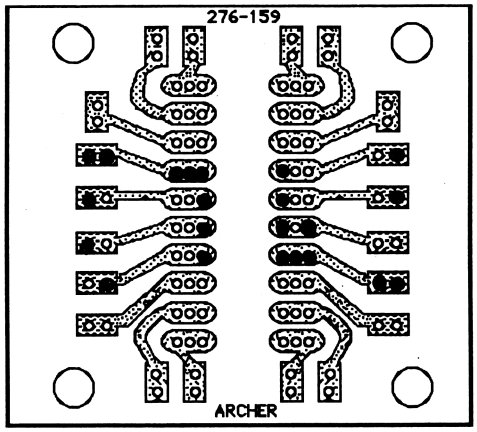


The fine wires in the cable are difficult to attach to the board, and it is easy to have a stray strand cause a short. We recommend you twist each wire and coat the wire with a thin layer of solder before soldering to the board.

☐ 7. **Attach the 16-conductor cable to the circuit board.** Solder +5V, PB2, AN2, AN3 and GND leads to the circuit board.

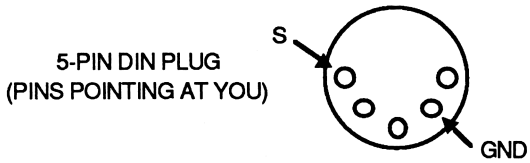


☐ 8. **Double-check all connections to the circuit board and insert the IC.** The copper-plated side of the completed board should look like the diagram below. The blackened holes are the only ones with wires in them. After the soldering has been checked, insert the LTC1092 IC in the chip socket. Be sure pin #1 (the pin to the left of the notch) is in the correct hole in the socket.



The Back of the Circuit Board After All the Components and Leads are Added

☐ 9. **Prepare the test-probe cable.** For the cable, you need 2 conductors ending in a 5-pin DIN plug.[2] A cable of this type is included in the Vernier Software parts kit. On the Vernier Software cable, the leads are color coded as follows: GND = black, S = red. If you need to build one, use a 5-pin DIN plug and 2-conductor wire. The two connections to use on the plug are shown in the diagram. If necessary, use an ohmmeter to identify the leads. The GND lead is connected to the ground of the power supply. The "S" lead will be used for the input signal. The voltage of this lead compared to ground will be measured by the A-to-D Converter.



Solder the test clips to the ends of the "S" and "GND" leads. To do this you will need to remove the tops of the test clips, slip them over the wires, solder to the metal inside the test clips, and replace the tops. If possible, use a red clip for the signal lead and a black clip for the ground lead. Label the two test clips and plug the test-probe cable into the DIN socket.

☐ 10. **Connect the unit to the game port and start up the computer.** Place the *Chaos in the Laboratory Programs #2* disk in the disk drive and start up your computer. If the computer does not start normally, turn it off and double-check your wiring. Otherwise, run the program A.TO.D. The computer should begin displaying the voltage.

☐ 11. **Measure the voltage of a flashlight battery.** Connect the two test leads across the terminals of a "C" or "D" battery. The positive test lead must contact the positive terminal of the battery. If the battery is fresh, you should read between 1.2 V and 1.7 V. Check the voltage using a voltmeter. The voltage displayed by the program should be steady and the same value as a voltmeter reads for the battery.

Disconnect the leads from the flashlight and touch them together to make sure the program displays zero volts. If the system is reading correctly at these two voltages (using a battery and 0 V), it is working. If the system is not reading correctly, refer to the "Troubleshooting" section below.

☐ 12. **Mount the DIN socket in the side of the box and put the lid on the box.** Two screws are used to secure the DIN connector to the side of the box. Use another screw to mount the

[2]Notice that only two of the three terminals that you connected on the DIN socket are used for this project. Other projects that use the A-to-D Converter will require the third (+5V) lead.

circuit board, upside down, to one of the posts in the corner of the box. Four screws are used to attach the lid on the box.

☐ 13. **Try the A.TO.D.AVG program.** This program adds two new features: (1) When the program starts, it asks you the reference voltage you are using. For the project you just built, the reference voltage comes from the Apple II's +5-volt power supply line. This voltage sometimes varies from 5 volts by a few tenths of a volt. For greatest accuracy, you can use a voltmeter to measure the reference voltage at pin #5 of the IC. Enter the reference voltage value to make the A-to-D Converter more accurate. (2) This program takes 64 voltage readings in quick succession and then calculates the average value. The results displayed by the program are these averaged voltages. This averaging tends to get rid of noise and produce more accurate readings.

☐ 14. **Try the SCOPE program.** The program SCOPE on the *Chaos in the Laboratory Programs #2* disk, used with the A-to-D Converter, will turn the Apple II into a storage oscilloscope. Try it out by typing RUN SCOPE and pressing <RETURN>. When asked about triggering, select N - NO TRIGGERING. If you have an oscillator or microphone, you might want to look at the signal using the oscilloscope. The "Project Extensions" section below includes other suggestions for projects using the SCOPE program. Included there are plans for building a microphone which will let you look at sound waves. Note that the vertical scale on the SCOPE program ranges from 0 to +5 volts with 2.5 volts at the center line. When using the SCOPE program you can press <H> at any time to hold the image. Press <H> again to return to normal oscilloscope operation. There are several sample screens in this manual that were made using the SCOPE program.

The triggering option of the SCOPE program allows you to specify a voltage (0 to 5 volts) for which the program will wait before starting a new trace on the screen. This is much like triggering on a real oscilloscope. Triggering can allow you to display steady waveforms on the screen.

If you are using an Apple IIGS, you may notice some unusual graphics effects when using the SCOPE program. The AN3 line that is used by the A-to-D Converter also switches the Apple IIGS from standard high-resolution graphics to double high-resolution graphics. This will cause a flicker of the screen if you have your computer set for monochrome display. To eliminate this problem, set the DISPLAY for COLOR using the Apple IIGS control panel.

Tips on Using the A-to-D Converter

- The voltage input range on the A-to-D Converter is limited to 0 to 5 volts. If the input voltage is outside this range, the display will read incorrectly. You may even damage the A-to-D converter with voltages outside this range.
- Do not attempt to directly measure the line voltage directly out of a wall socket, as the AC voltage varies up to ±190 volts!
- All voltage measurements made by the A-to-D Converter are with respect to the Apple's power supply ground lead. The GND test lead from the A-to-D Converter is a grounded wire.[3]
- Be aware of polarity of the leads. The "S" test lead must always be more positive than the GND lead.
- Use the A.TO.D program for steady or slowly changing voltages (as a replacement for a voltmeter).
- Use the SCOPE program for rapidly changing voltages (as a replacement for an oscilloscope). Note that the voltage range of the scope is 0 to 5 volts, so the midpoint of the voltage range is 2.5 volts, not 0 volts as it is on a standard oscilloscope. Some microphone and oscillator circuits will need offset voltages added to display correctly on the A-to-D Converter oscilloscope (see "Project Extensions").

[3]One of the project extensions shows how to modify the circuit to measure differential voltages.

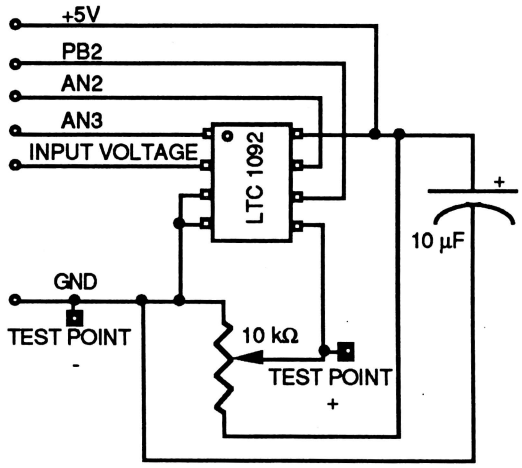
Troubleshooting

◊ Check your connections on the circuit board and cable by using an ohmmeter. Make sure pins #3 and #4 are both connected to GND. Is your capacitor connected across the proper pins of the IC? The negative lead of the capacitor should have a direct connection with pin #4 and the positive lead of the capacitor should have a direct connection with pin #8 of the IC.
◊ Be sure there are no solder bridges on the circuit board. Check this with the chip removed from the socket and the cable disconnected.
◊ Check the 5-pin DIN connections inside the box. Refer to the diagram in step 5 of the "Step-by-Step Instructions."
◊ Check the ribbon cable connections inside the box. Refer to the diagram in steps 6 and 7 of the "Step-by-Step Instructions." If any one of the five connections is wrong, the circuit will not work.
◊ If you are using an Apple IIe or IIGS, make sure nothing is plugged into the 9-pin game port.
◊ Connect the DIN cable and check again. Be sure to mark GND and +5V on the test leads.
◊ Try reheating the solder joints on the board.
◊ Make sure that you connected the two jumper wires correctly.
◊ As a last-ditch effort, try a different chip. The LTC1092 can be damaged by voltage spikes or static discharge. A damaged chip may still respond but will give nonlinear results.

Project Extensions

Circuit Modifications

◊ Add a voltage range adjustment. One of the most powerful features of the LTC1092 chip is that you can set the range of voltages measured by controlling the voltage applied to pin #5. In the core project, we applied +5V to pin #5 so that the range of the circuit was 0 to 5 volts. By reducing the voltage at pin #5, you can set a smaller range. Note that this makes the circuit more sensitive. It has the same effect as adding an amplifier to a traditional A-to-D converter. The schematic below shows a 10-k potentiometer added to the circuit to allow you to control the reference voltage. It also indicates two test points where you can connect a hand-held voltmeter when setting the reference voltage.



A-to-D Converter With Voltage Range Adjustment

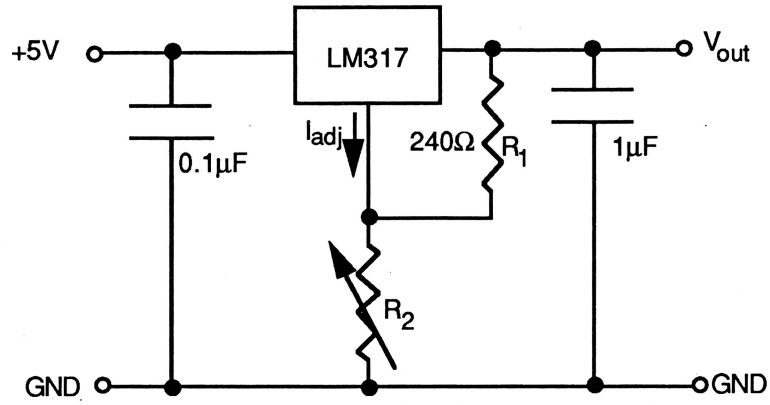
The count displayed by the A.TO.D program will always be:

$$\text{count} = \frac{\text{Voltage In}}{\text{Reference Voltage}} * 1024$$

If you use any reference voltage other than 5 volts, the programs A.TO.D and SCOPE will not be calibrated correctly, because they assume the 5-volt reference. The program A.TO.D.AVG allows

you to specify the reference voltage. Use it if you have a reference voltage different from 5 volts. Do not try to use a reference voltage higher than 5 volts.

◊ Add a voltage reference IC to the circuit. Using either the core project or the adjustable reference voltage circuit described above, the reference voltage depends on the Apple II power supply voltage. The "+5V" output of this power supply can vary a little. For the greatest accuracy of voltage measurement, you can add a voltage reference IC to the circuit. This allows you to move the unit from one computer to another without the need to recalibrate. Either a zener diode circuit or an adjustable voltage regulator (LM317) would be suitable. The circuit below shows how to use an LM317 to provide an adjustable voltage reference. The reference voltage out of this circuit is given by the equation: $V_{out} = 1.25 * (1 + R_2/R_1) + I_{adj} * R_2$



A Circuit to Provide a Variable Reference Voltage

With the LM317, you could even make a software programmable reference voltage by using the other two annunciator lines on the Apple II to operate transistor switches that vary the resistor (R_2) in the voltage regulator circuit.

◊ The LTC1092 has differential inputs. The voltage measured is the difference in voltage between pins #3 and #2. In the circuits above we have pin #3 connected to the computer's power supply ground, so that all voltages measured are relative to ground. If you disconnect pin #3 from ground (pin #4) and use a separate negative lead from pin #3, your circuit can measure floating voltages.

◊ Since each LTC1092 ADC requires two annunciators and the Apple II has four available, it should be possible to develop a two-channel ADC. This would require two LTC1092 chips. The second chip could be controlled by AN0 (clock) and AN1 (chip select). The serial data would flow into the computer via PB0. The machine-language routine would have to be modified to use both chips. In effect, two voltages could be monitored at once.

◊ Linear Technology Corporation makes a 2-channel, 10-bit serial output ADC (LTC1091) in an 8-pin package. It should be possible to control this chip using three annunciators. The extra annunciator would be used to select the channel. If you had a project where you needed to measure two voltages, and you also needed an annunciator to control another device (valve, solid state relay, LED, etc.), then this chip would be worth investigating. The machine-language routine for controlling this chip would require an experienced programmer.

Program Improvements

The BASIC programs A.TO.D and A.TO.D.AVG are fairly simple programs that can be improved by adding new features. The "Program Design Notes" section below explains how the programs operate. It should help you in making modifications. Some suggested changes are given below:

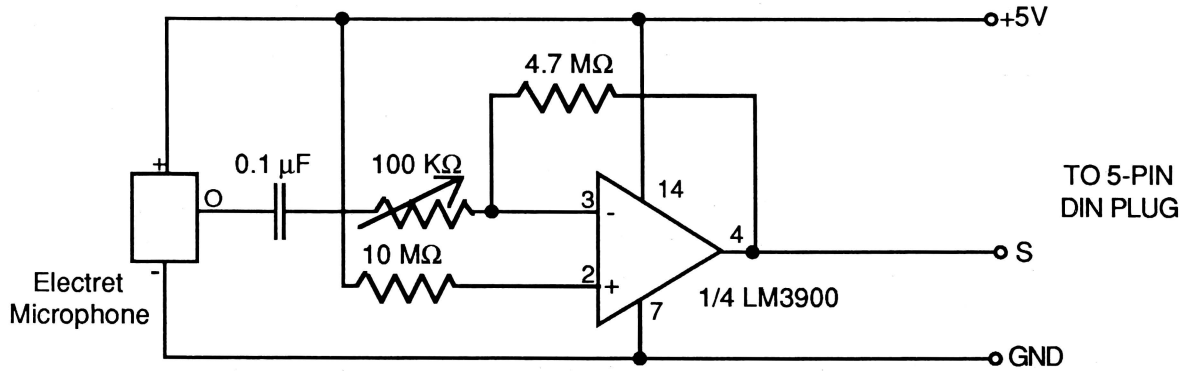
◊ Modify the programs to print the data table on your printer.
◊ Allow the voltage and time data to be saved on a disk and reloaded later.

◊ If your computer has a "clock card" to keep track of real time, change the programs so that they read the clock time and store it along with the voltage reading.
◊ Add a mode to the programs that graphs the voltage readings versus time.
◊ Add a mode to the programs that displays the voltage in large low-resolution graphics block digits so that they can be read at a distance.
◊ Modify the machine-language routine so that it stores many voltage readings in a block of memory. These values can later be examined and displayed using a BASIC program.
◊ For many applications, an 8-bit ADC would be satisfactory. Modify the machine-language routine LTC1092 so that the two least-significant bits of data are discarded. This would mean that one byte would be required to store each data point in memory (more data in less memory), and the program would also be faster. The project requires machine-language programming.
◊ If you are using the circuit with a transducer, the program A.TO.D could be modified to display units of pressure, temperature, light intensity, or whatever your transducer provides.

Additional Experiments

Since this project gives you a voltmeter/oscilloscope, there are many experiments that can be done. Four of the other projects in this book require a voltage measurement device such as the A-to-D Converter: Light Meter, Barometer, Magnetic Field Sensor, and Smart Battery Charger. The book *How to Build A Better Mousetrap* describes three more projects that can plug directly into this unit: pH Amplifier, Strain Gage, and Thermocouple. Some additional suggestions are listed below. Many of them take advantage of the fact that the A-to-D Converter can read voltages very rapidly.

◊ The SCOPE program is good for displaying sound waves. A microphone/amplifier combination, with its output centered at 2.5 volts, can be used to pick up the sound. An inexpensive microphone circuit is shown below. The microphone (Radio Shack #270-092) sends a small voltage signal to the amplifier. The 100-kΩ pot adjusts the sensitivity of the amplifier. The circuit should be built with a 5-pin DIN plug on the end so that it will plug directly into the A-to-D Converter. You can hum, sing, or play instruments into the microphone as a source of sound. A recorder (the musical instrument) will give notes that appear as sine waves.



A Microphone/Amplifier Circuit For Use with the A-to-D Converter

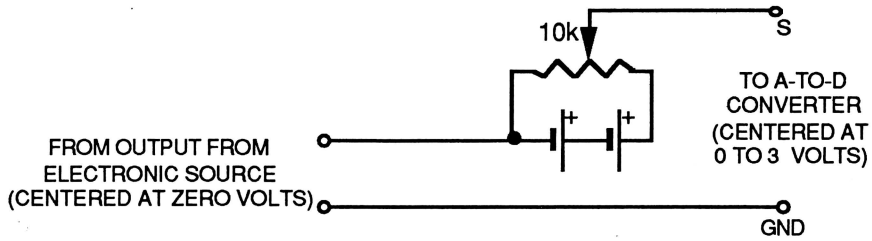
This microphone can also be used with Project 10 of this book to digitize sounds. This microphone is available from Vernier Software either assembled (mounted in a project box), or as a parts kit (with no project box). Both of these use the 5-pin DIN plug so they will plug into the A-to-D Converter. The order codes and prices are:

Microphone/Amplifier Parts Kit..... Order Code: MIC-U $15.00
Microphone/Amplifier Assembled..... Order Code: MCA-U $30.00

We sell several other similar microphone/amplifiers, so be sure to specify the order code when ordering.

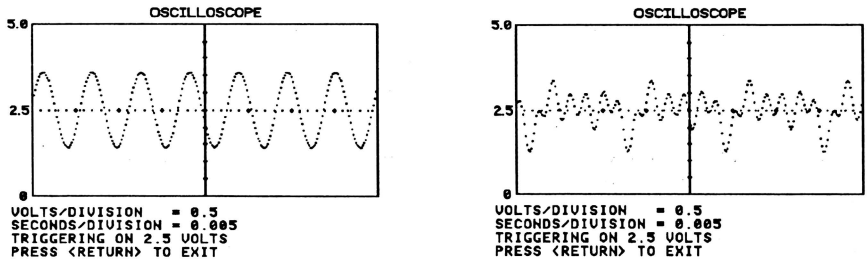
◊ A tape recorder, function generator, or other electronic device can also be used as an input. Electronic synthesizers or keyboards provide a great, inexpensive signal source for experimenting with sound waves and the oscilloscope. Most of the synthesizers have an "output" connector for hooking up earphones or external speakers. You can connect the A-to-D Converter to this connector.

In most situations in which an electronic signal is connected to the A-to-D Converter, you will need to add a circuit to offset the voltage so that it is centered at 2.5 volts rather than 0 volts. If you just connect directly to the A-to-D Converter, you will get only the positive half of the waveform displayed on the oscilloscope screen. The circuit below solves this problem. It provides an offset voltage using two 1.5-volt batteries. Adjust the potentiometer until the waveform is centered on the scope screen.



A Voltage Offset Circuit

The sample screens below show SCOPE screens made using signals from a Casio keyboard. The one on the left shows a flute wave pattern and the one on the right shows a jazz organ wave pattern. Since the sampling rate of the A-to-D Converter is limited to about 6000 samples per second, the program will work best when you use the lower notes on the keyboard. At frequencies above about 1000 hertz, the wave pattern will be difficult to interpret.



Sample SCOPE Screens Made Using a Microphone

◊ The A-to-D Converter can be used to digitize sounds which can then be played back using the D-to-A Converter / Digitized Sound project (Project 10). Refer to the "Project Extensions" section of Project 10 for details. All of the digitized sounds on the *Chaos in the Laboratory Programs* disks were sampled using this A-to-D Converter.

◊ The A-to-D Converter combined with a pressure transducer could be used to study rapid pressure changes. Attach the pressure transducer tube into the neck of an inflated balloon. Record the pressure changes as you squeeze, punch, and eventually pop the balloon.

◊ Make a series of homemade batteries and measure the voltage of each. Suggestions for batteries are given below:

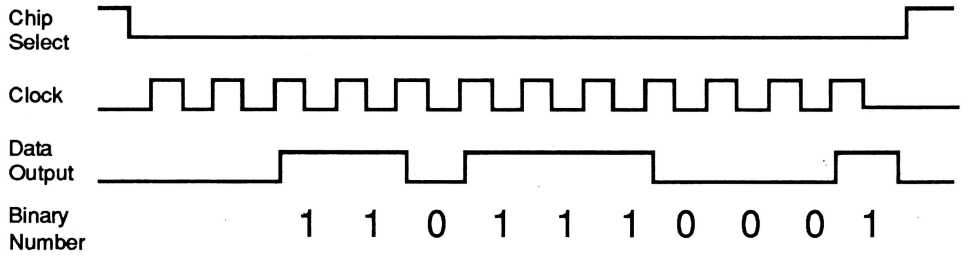
- Lemon (or other citrus fruit) with a copper wire stuck in one end and an uncoated nail (finishing nail) in the other end.
- A nickel, paper towel, penny sandwich. Soak the towel in vinegar.
- Aluminum foil and copper wire in a dilute baking soda solution. The wire should not touch the foil.
- Zinc (galvanized steel) and copper wire in a solution of salt water.

◊ The junction between two different metals forms a thermocouple (Project 8 in *How to Build a Better Mousetrap*). The voltage of the junction is proportional to the temperature. Smashing a piece of metal with a hammer causes a rapid increase in the metal's temperature. Cross two metal wires (different materials) over a smooth hard rock (nonconductive surface). Have one end of one wire connected to the A-to-D Converter GND lead and one end of the other wire connected to the V_{in} lead. Run the SCOPE program to collect data quickly, and hit the wire intersection with a hammer. If no voltages are observed, reverse the wires and try again, using a new portion of the wire. The voltage change is small, so you may need to amplify the signal or set the reference voltage to a lower level. **Note: You should wear safety glasses since pieces of wire or rock could fly out.**

Supplemental Information

How the Voltage is Read from the LTC1092

As mentioned in the "Background Information," the LTC1092 has serial output. This means that it transmits its voltage readings to the computer as a series of 10 high- or low-voltage signals. The computer must be able to interpret this stream of information. A short machine-language routine decodes the data into two 8-bit bytes that the computer can manipulate. The computer controls the LTC1092 using two annunciator output lines and it reads the data over a digital input line. When the computer wants the ADC to take a voltage reading, it switches the chip select line (AN3) to low. This starts the analog-to-digital conversion. The AN2 annunciator line from the computer that is connected to the clock pin is now put through a sequence of 12 low-to-high transitions. After the third low-to-high transition, data transmission begins.[4] Each time the clock line changes from low to high, the data output line is set for a different bit of analog-to-digital conversion result. The most significant bit is acquired and transmitted first. That binary number is available at the output terminal until the next time the clock line goes high. At that point the next data bit is acquired and available at the output terminal. Each time the clock goes high, another data bit is sent to the output line. After 12 clock cycles (2 start bits + 10 data bits), the ADC is ready for the next conversion. The machine-language routine rolls the bits one at a time into the computer's accumulator register. After the first two bits are in the accumulator, that data is stored and the process continues for eight more bits. The first two bits are just rolled off the end of the 8-bit word. After eight bits, the data in the accumulator is stored again. The Apple II computer uses 8-bit words, so the first two bits go into one word and the other eight bits go into the second word. Six bits of the first word are wasted.[5] The chart below shows the sequence of signals which result in a single 10-bit count being read by the computer.



Machine-Language Routines

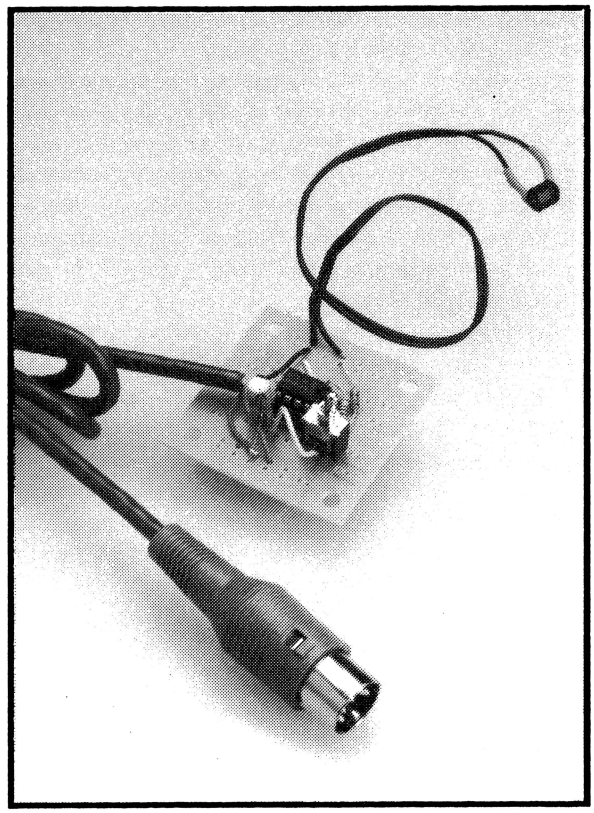
If you have any familiarity with Apple II machine language, or if you want to learn a little about it, you might want to examine the code that reads the data from the LTC1092. Assembly-language source files A.TO.D.SRC and A.TO.D.AVG.SRC are included on the *Chaos in the Laboratory Programs #2* disk. Refer to *Appendix E* for information on how to view, modify, and compile these files. You can also look at this code on the computer's monitor. At the Applesoft prompt "]" type in BLOAD

[4]The first two clock cycles are called *start bits*. You may have heard of start bits if you have used a modem for telecommunications.

[5]One of the machine-language routines (A.TO.D.AVG.0) takes advantage of the "wasted" bits for signal averaging.

A.TO.D.0, A$300 and a <RETURN>. After the disk stops, type in CALL -151 and the computer comes back with an "*" instead of the usual "]". Next type in 302L and a <RETURN>.[6] This will list 20 lines of the code in a disassembled form. To list more, just use "L" followed by a <RETURN>. To return to Applesoft BASIC use <CONTROL> <C> followed by a <RETURN>.
Three infrequently used machine-language instructions are used in this program: BIT, ASL and ROL. The BIT command does a Boolean logic AND between the accumulator and the memory location indicated, but it changes neither the value in the accumulator nor the value in memory. In our case it is used because it is fast and it does not change the accumulator. BIT $C05C turns on annunciator 2 (AN2) and BIT $C05D turns off annunciator 2. This annunciator serves as the clock for the ADC. BIT $C05E and BIT $C05F turn on and off annunciator 3, which serves as the chip select line for the ADC. ASL is arithmetic shift left and is usually used in multiplication functions in the computer. Here we use ASL $C063, which is the memory-mapped location of the PB2 digital input. The PB2 line is controlled by the output line of the LTC1092. If the PB2 line is at a high voltage, the seventh bit of this byte is high, and the ASL operation causes the carry flag on the status register to be set. If the line is low, the seventh bit is low and the carry flag is not set. The ROL is called to rotate the bits left through the carry bit. The binary number in the accumulator is shifted one bit to the left and the carry is shifted into the right end. The last data bit is detected by the ASL $C063 and the carry flag is set, then the ROL moves that bit into the accumulator. When the next data bit comes along, the first data bit is shifted left and the new data bit from the carry flag is shifted in.

[6]Since the program was loaded at address $300 (300 in hexadecimal notation), you may wonder why we don't use 300L to list all of the program starting at $300. The data read by the routine is stored in the first two bytes of the code.



Project 6

Light Meter

This project is a circuit that will allow you to measure light levels using the computer. A silicon photodiode is used as the sensor. The photodiode produces a small current when it is exposed to light. This current increases in proportion to the illuminance. We will build a simple op-amp circuit to produce an output voltage that is proportional to this current. Several different voltage input devices can be used to read this voltage into the computer. You can use this project for performing a variety of experiments with light in the laboratory or for solar energy studies.

Project Requirements

Voltage Input Device

You will need one of the following voltage input devices:

- A-to-D Converter from Project 5 of this book
- Voltage Monitor from Project 7 of *How to Build a Better Mousetrap*
- Vernier Software 16-pin Apple II Voltage Input Unit
- Vernier Software 9-pin Apple II Voltage Input Unit (requires a one-line program change—see *Appendix F – Teacher's Guide*)

This project can also be used with other software and any of the following laboratory interfaces. All of these devices accept the 5-pin DIN plug used for this project.

- Vernier Software Voltage Input Unit (Apple II or IBM-compatible)
- Vernier Software MultiPurpose Lab Interface (Apple II or IBM-compatible)
- Universal Lab Interface (Macintosh, or any computer with a RS-232 port)

Computer

This project can be done using any Apple II computer, providing the appropriate voltage input device is used.

Background Knowledge

- Electronics: This is a fairly easy project. Some soldering experience is recommended. Knowledge of Ohm's law and some familiarity with electronics terminology are also helpful.

- Programming: None required for the core project. Some of the project extensions require experience with BASIC programming.

Parts

- ☐ circuit board (1/2 of a Radio Shack #276-159) (If any other circuit board is substituted, the layout will have to be changed, making the project a little more difficult.)
- ☐ silicon photodiode (Hamamatsu S1133 or equivalent, Hamamatsu Corporation, P.O. Box 6910, Bridgewater, NJ 08807, (908) 231-0960).
- ☐ CA3140AE op-amp (Mouser Electronics #570-CA3140AE)
- ☐ 8-pin socket (Radio Shack #276-1995)
- ☐ 10-μF electrolytic capacitor (Radio Shack #272-1025)
- ☐ 1-MΩ, 1/4 watt, 1%, metal-film resistor (brown, black, black, yellow, brown)
- ☐ 100-kΩ, 1/4 watt, 1%, metal-film resistor (brown, black, black, orange, brown)
- ☐ 4.02-kΩ, 1/4 watt, 1%, metal-film resistor (yellow, black, red, brown, brown)
- ☐ switch (SPDT, i.e., single-pole, double-throw ON-OFF-ON). Mouser Electronics #MS-10TC612 or equivalent
- ☐ cable with 5-pin DIN plug for connection to voltage input device
- ☐ heat-shrink tubing (4-cm) or electrical tape
- ☐ (2) jumper wires for connection on the circuit board
- ☐ 2-conductor wire (about 20 cm)

Note: A Light Meter Parts Kit containing all of the components needed for this project is available from Vernier Software. This kit sells for $16.00 (order code CLM-DIN). An order form is provided at the back of this book.

Tools and Supplies

Low-power soldering iron or soldering pencil, solder, wire cutters, wire strippers, and a multimeter (DC voltmeter and ohmmeter).

Background Information

In this project you will build a circuit to measure light intensity and use it with a computer. But what exactly is light intensity and how is it measured?

Light intensity has traditionally had many different units which are sometimes very confusing. Some examples that have been used in the past include brils, nox, stilbs, blondels, glims, Hefnerkerze, phots, scots, helios, lumergs, pharos, fluorens, and Trolands!

What we are measuring here is technically called *illuminance*. Illuminance is defined as the luminous flux incident per unit area, where *luminous flux* is a measure of the light energy passing through space. In common terms, you might say illuminance is a measure of how much light is striking a surface per unit area, or "how bright it seems to our eyes." Illuminance is measured in *lumens* per square meter, or *lux*. There is also the English unit for illuminance called a *foot-candle*, which is a lumen per square foot. One foot-candle is equivalent to about 10.76 lux.[1] One nice thing about the foot-candle unit is that it is relatively easy to understand. A foot candle was originally defined as the illuminance on a surface one foot away from a standard candle. In fact, at one time, there actually was a *British standard candle,* which was designed to burn at some reasonably constant rate. A lumen was defined as the luminous flux emerging through an area of one square foot at a distance of one foot from this British standard candle. Today there is a more arbitrary, but also more reproducible metric standard. In any case, what is important for us to know is that the unit lux (lumens per square meter) exists and we can make measurements and comparisons with it, just as we can for distance using meters, centimeters, feet, etc.

Here is a table giving typical illuminance values in lux of differing sources and conditions:[2]

Situation	Illuminance (Lux)
Brilliant day, sun overhead	100,000
Overcast day	8,000
Minimum recommended for fine work	500
Night football or baseball	400
Office, classrooms, reading room	300
Street lights	5
Full moonlight	0.2
Starlight	0.0003

Illuminance measurements are *photometric*; that is, they measure light levels as they are perceived by the human eye. The eye has different sensitivities to different kinds of light. A source of green light at some given energy will seem brighter to the human eye than a source of red light at the same energy. Illuminance measurements take this into account. Other measurements of light intensity are *radiometric*. Radiometric measurements take into account all of the energy in the light over the whole electromagnetic spectrum. For most light, there is quite a lot of energy in the infrared wavelengths. Infrared radiation is considered in radiometric light measurements, but not in photometric measurements. One of the project extensions describes a radiometric light meter.

There are many different electronic devices that can be used to measure light intensity. Some of the most common are:

Photoresistors: These are small, light-dependent resistors usually made of cadmium sulfide (CdS) or cadmium selenide (CdSe). These sensors are inexpensive and readily available. Their spectral response nicely matches the human eye.

Photodiodes: Photodiodes are usually made of silicon. Photons of light striking the silicon knock loose electrons, which then move through the silicon.

Phototransistors: Phototransistors are similar to normal transistors, except that a light-sensitive region of the transistor can produce a small current from the base to the emitter when light strikes it.

---

[1]This conversion comes from the fact that 1 m^2 = $(3.28\ ft)^2$ = 10.76 ft^2. So 1 foot-candle = 1 lumen/ft^2 = 1 lumen/((1/10.76) m^2) = 10.76 lumen/m^2 = 10.76 lux.

[2]*Elements of Physics*, John N. Cooper and Alpheus W. Smith, McGraw-Hill, 1973, p.381.

As in any transistor, a small base-to-emitter current can cause a large collector-to-emitter current if the transistor is wired correctly.

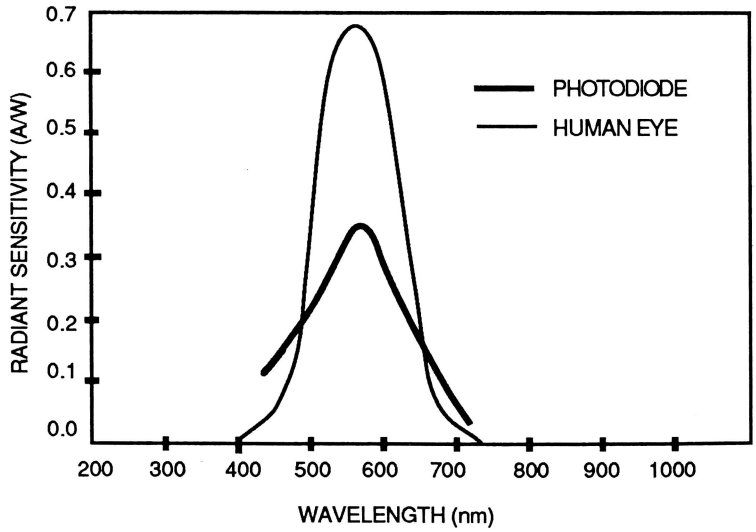
Phototubes and photomultipliers: These are metal or glass tubes with most of the air pumped out of them. A metal rod down the center of the tube is at a high voltage relative to the metal can itself. When a photon of light enters the tube it can knock loose electrons and cause ions to be formed. A high voltage across the tube causes the charged particles to move, producing a current. These devices can be very sensitive to light, but they are expensive and require a high-voltage power supply.

If your goal is to make a simple, inexpensive light sensor, photoresistors may be what you want to use. A photoresistor circuit is described in the *Introduction* of this book. However, photoresistors have several characteristics that would limit the accuracy of a photoresistor-based light meter:

- Photoresistors become less sensitive with time, especially if they are exposed to ultraviolet light, high temperatures, or high humidity.
- Photoresistors often change their response depending on the light level they were exposed to previously. For example, a photoresistor that was stored in the dark will be slightly more sensitive than an identical photoresistor that has been exposed to bright light for several minutes.
- The sensitivity of photoresistors change noticeably with temperature.
- Photoresistors have a slow response time; that is, it takes them a tenth of a second or so to respond. This is not important for most experiments, but in some cases (for example, if you are trying to show the flicker of a light source) it may cause problems.
- It is useful only over a narrow range of light levels.
- The photoresistor does not respond in a linear way to changes in illuminance.

Because of these limitations, we are going to use a little more complicated, but considerably improved circuit for our light meter. First, we are going to use a silicon photodiode instead of a photoresistor. The photodiode produces a small current proportional to the illuminance. Then we are going to connect the photodiode into a simple op-amp circuit to produce a voltage that is proportional to illuminance. Finally, since the Apple II game port cannot read voltages directly, we will have to use a device to read the voltage into the computer. The A-to-D Converter (Project 5) of this book or several other voltage input devices can be used.

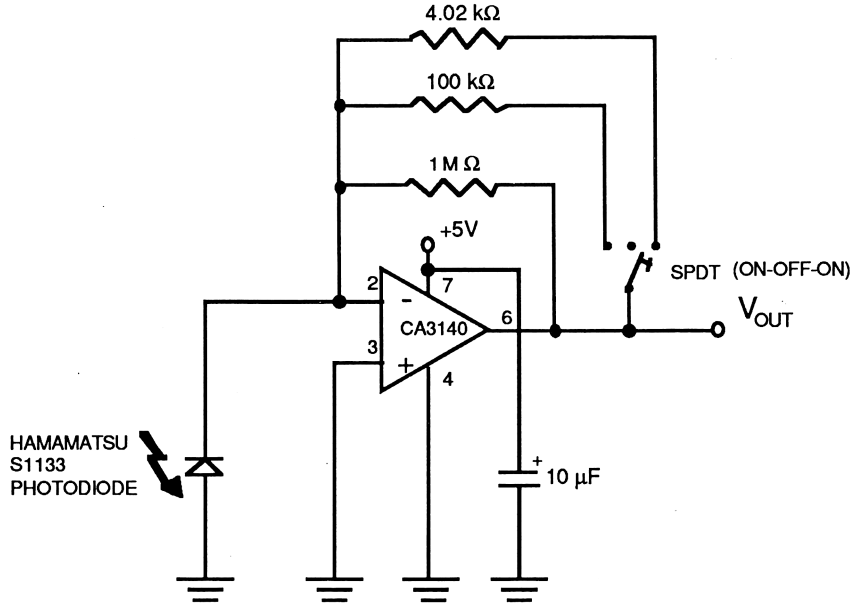
A photodiode is sensitive to both visible and infrared light, but the sensor we are using for this project (Hamamatsu S1133) has a filter built into it so that infrared light is rejected. The spectral response of the sensor, with the filter, closely matches the response of the human eye. This graph shows how the sensitivity of the photodiode changes with wavelength.[3] It also shows the relative spectral response of the human eye (here the shape of the curve is important and not the units).



Spectral Response of the Hamamatsu S1133 Photodiode

[3]The units of the sensitivity of the diode are given in A/W, which is amperes per watt. This is basically how many amperes of current the photodiode produces for every watt of incoming light power.

The complete photodiode light meter circuit is shown below:



Light Meter Schematic

Here is a brief explanation of how the circuit works. Photodiodes are made up of two slightly different types of silicon which are joined together in the center of the diode. When light strikes the silicon near this junction, electrons are excited and a net migration of charge occurs. If the photodiode is connected to a circuit like this one, these electrons migrate toward the cathode end of the photodiode (marked with a bar on the schematic), producing a small current. This current is directly proportional to the illuminance of the photodiode. If the illuminance doubles, the photodiode current doubles.

The op-amp's job is to produce a voltage which is proportional to this tiny current from the photodiode. Using the principle of negative feedback, the op-amp produces a voltage equal to the product of the photodiode current and the resistance of the feedback resistor.[4] The three feedback resistors are selected by the setting of the SPDT switch, which gives us different ranges for the meter. If the switch is connected to the 100-kΩ resistor, then the range of the meter is about 0 – 6000 lux. This gives you a good, general-purpose "light meter." If the switch is in the "off" position, then the range of the meter is a more sensitive 0 – 600 lux. This scale is best used only for experiments involving very low light levels. If the switch is connected to the 4.02-kΩ resistor, the range of the meter is 0 – 150,000 lux. This should be useful only for measurements involving sunlight (see "Additional Experiments"). The programs on the *Chaos in the Laboratory Programs #2* disk are designed for use with this circuit. They allow you to specify which resistor is being used by pressing the <SPACE BAR>.

You can use this Light Meter for a number of interesting experiments. First, you may want to simply determine the illuminance in areas around your house or school. You may also want to try to experimentally verify the inverse square law. In theory, illuminance from a point source of light should be inversely proportional to the square of the distance from the source. You can test this concept with the Light Meter circuit. Other experiments include the study of interference and diffractions patterns, studies in solar energy, measurement of the flicker rate of fluorescent light, and using the circuit as part of a system to control lamps as light level changes. Several of these experiments are discussed in the "Project Extensions" section.

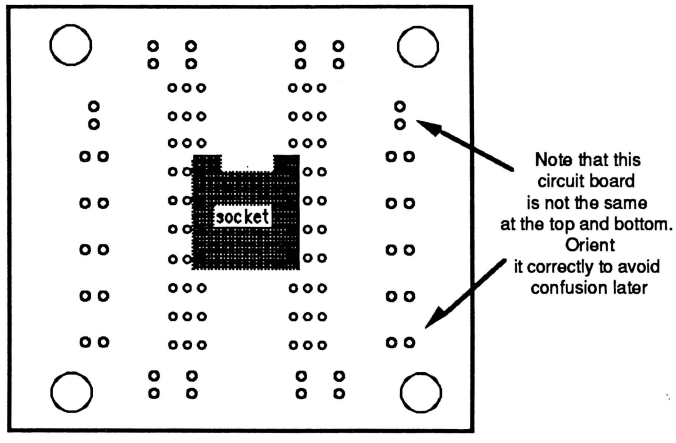
[4]Refer to the "Supplemental Information" section for a more complete discussion of how this circuit works.

Step-by-Step Instructions

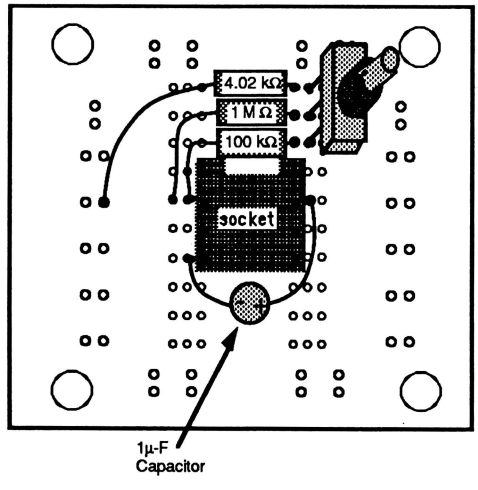
These instructions assume that you are using one-half of a Radio Shack #276-159 circuit board. If you are using a different circuit board, the layout of parts will have to be changed.

☐ 1. **Identify all of the components.** It is important to make sure that you have all the materials for this project before you begin. Double-check the parts list given at the beginning of this project, and identify each resistor and capacitor.

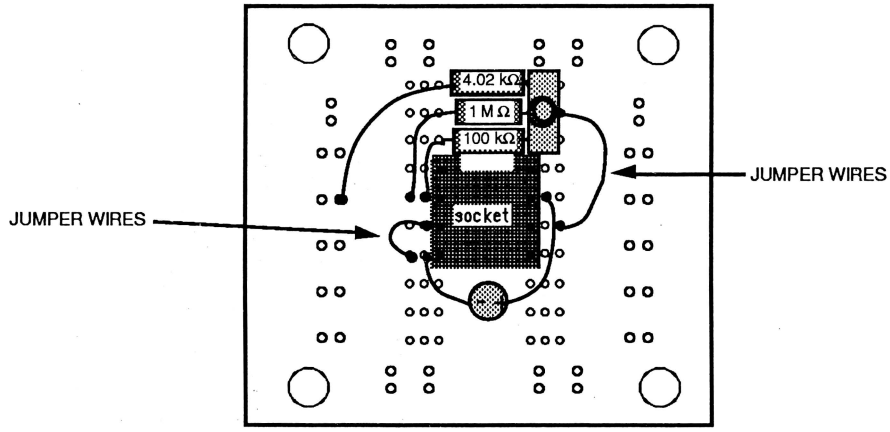
☐ 2. **Place the 8-pin socket on the circuit board.** Hold the circuit board with the plain side up (no copper plating or writing). Place the socket on the circuit board, on the unplated side of the circuit board, in the position shown in the diagram below. The top of the socket will be marked with a notch or other mark to indicate pin #1. The pins are numbered counter-clockwise (as viewed from above) from pin 1, on the upper left corner, around to pin 8. Solder all 8 of the pins to the plated side of the circuit board. Do not place the op-amp IC into the socket at this time.



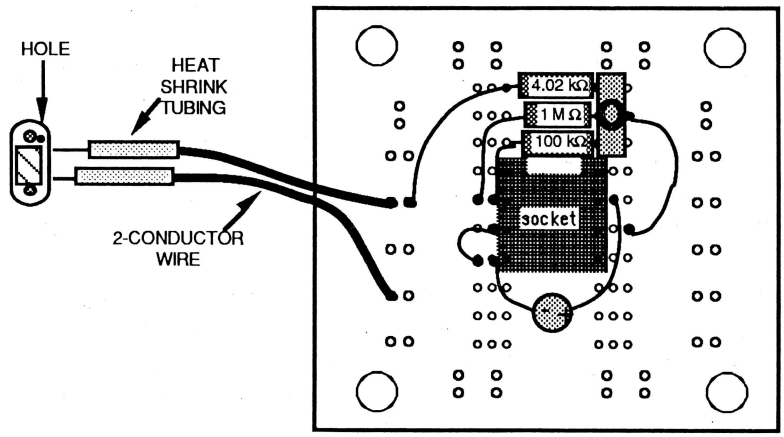
☐ 3. **Solder the resistors, switch, and capacitor in place.** Position the three resistors, switch, and capacitor as shown in the figure below. Be sure to orient the electrolytic capacitor in the proper direction. Note that one of the leads of the capacitor is marked with a "–" on the case. Put each component into the board as far as it can go with the leads all the way through the plated side. You can bend the leads on the plated side so that the components stay in place while you solder them.



☐ 4. **Add the two jumper wires.** Solder a jumper wire from pin #3 to pin #4 and another wire from pin #6 to the switch as shown in the diagram below:

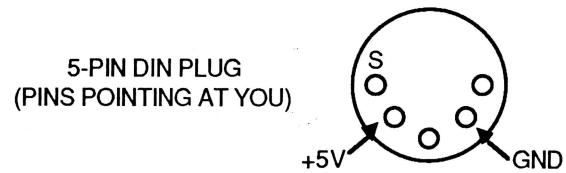


☐ 5. **Connect the photodiode.** The photodiode is connected to the circuit board using a 2-conductor cable. This wire should be kept short (20 cm or so) to minimize electrical noise.[5] The orientation of the photodiode is shown in the diagram below. The side with the small hole should be connected to pin #2 of the IC socket. Strip a few mm of insulation from both ends of both wires. Place short sections of heat-shrink tubing over the wires and then solder them to the photodiode. Keep your soldering time as short as possible, as excessive heat may damage the photodiode. Warm the heat-shrink tubing with your soldering iron to tighten it around the solder joint. Connect the wires leading from the photodiode to the circuit board as shown below.



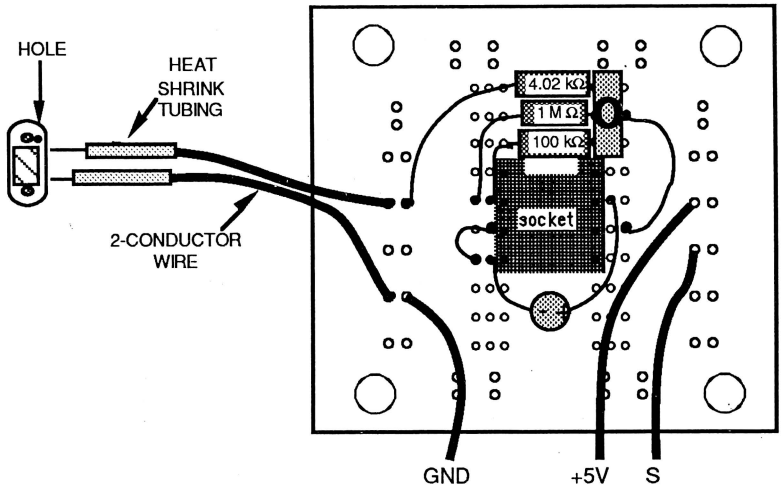
☐ 6. **Connect the DIN cable.** This circuit connects to the voltage input device using a 3-conductor cable with a DIN connector on the end. The three connections to use on the cord are shown below. In Vernier Software parts kits, the following color code is used to identify these three leads: black = GND, red = signal, orange = +5V.

If you are using your own cable, use an ohmmeter to identify which wire on your cable connects to each terminal on the plug.



Solder the three leads to the circuit board as indicated in the diagram below:

[5]If you want to have longer wires on your photodiode, refer to the "Project Extensions" later in this chapter.



DIN Cable Connections

☐ 7. **Trim the leads** on the plated side of the circuit and check all connections. Check for any solder bridges between places that should not be connected.

☐ 8. **Plug in the IC.** Carefully place the op-amp into the socket. Be sure to orient pin #1 at the proper corner of the socket (to the left of the notch). Make sure all eight pins go into the socket.

☐ 9. **Test the completed circuit.** Turn off the computer. Connect the voltage input device to the computer and plug in the Light Meter circuit. Set the switch to the up position. Place the *Chaos in the Laboratory Programs #2* disk in the disk drive and turn on the computer. Run the program LIGHT. If you are using the A-to-D Converter (Project 5 of this book), enter "A" when asked which type voltage input device you are using. If you are using a Voltage Monitor (from *How to Build a Better Mousetrap)* or a VIU enter "V". The computer should begin displaying the illuminance reading.

☐ 10. **Adjust the circuit for the kinds of measurements you plan to do.** Changing the position of the switch allows you to change the range of the meter. With the SPDT switch in the up position (away from the IC), the Light Meter is on the medium range (0 to about 6000 lux). This is the setting you will use for most tests and experiments. However, if you need to measure lower illuminance levels, you may wish to use the lower range. By placing the switch in the middle position, the Light Meter will be in the range from 0 to about 600 lux. With the SPDT switch in the down position, the Light Meter circuit is on the high range 0 to 150,000 lux. This range should be used only in sunlight. Note: You must also change the calibration in the program when you change the switch setting. This is done by pressing the <SPACE BAR>. The program will display the range on the monitor. It might also be helpful to write or label the three switch positions on the circuit board for the medium (up), low (middle), and high (down) ranges.

Troubleshooting

If your circuit does not behave as you expect it to, check the following:

◊ Is your IC inserted properly? Pin #1 is marked with a dot. Is it where it should be?
◊ Have you bent one or more pins on the IC by not putting it into the socket properly?
◊ Does your op-amp get hot when you turn things on? If it is hot, you must have a serious wiring problem, and you may even need to replace the chip after you fix the wiring error.
◊ Do you have the photodiode wired correctly? The end with the small hole should be connected to pin #2 of the op-amp. The other end should connect to pin #4.
◊ Double-check the DIN connections with an ohmmeter.
◊ Make sure you are using the correct resistors. The parts list at the beginning of this chapter gives color codes to help in identifying the resistors.

◊ Visually double-check all of the connections on the op-amp. If you cannot see anything in the wrong place, check again with the ohmmeter between the IC pins and the connections to the various components.
◊ Are there any solder bridges on the plated side of the board connecting two places on the circuit that should not be connected? Are the pins on the IC all soldered properly, and not just sticking through the board?
◊ Is the switch in the right position? For most experiments, you will want the switch in the up position.
◊ Try reheating all the solder joints. Sometimes this will fix an unseen cold solder joint.

If none of these problems are found, it is time to check the voltages on the various pins of the op-amp IC with a voltmeter. The negative lead of the voltmeter should be connected to the GND line (pin #4 on the IC). These tests should be done with the circuit connected to the voltage input device and the program running.

Pin #	Voltage Should Be	Comments
7	+5 V	If not +5 V, check the DIN cable connections.
3	0 V	If not 0 V, check the jumper wire connecting pins #3 and #4.
2	very close to 0 V	This is the signal from the photodiode, but a wiring error in the op-amp circuit or a bad IC could cause a problem here.
6	0 to 3 V	This is the output voltage and it should read the same voltage that the program displays. It should change as you change the light level. If this voltage is ok, your problem must be in your voltage input device or its connections. If the voltage at pin #6 does change, then the problem will most likely be in the voltage input device or its connections. Check the voltage input device by applying a known voltage signal.

Project Extensions

Circuit Modifications

ADD A LONG CABLE FOR THE PHOTODIODE

In some cases, you may want a long cable on the photodiode so that you can move it around easily. This is not a problem, except that long wires on the photodiode can act as an antenna and pick up electrical noise that is then amplified along with the signal from the photodiode. If you use coaxial cable for this wire, the electrical noise will be minimized. A coaxial cable has a metal outer shield completely enclosing a second wire. The outer shield should be connected to ground. The inner wire is used for the signal from the photodiode to pin #2.

ELIMINATING OSCILLATIONS

The Light Meter will respond very quickly to changes in light level; in fact, a project in the "Additional Experiments" section uses the Light Meter to study the flicker of fluorescent lamps. There are other situations where you may want to "smooth out" the signal from the Light Meter. To average out the variations in the signal caused by the AC voltage, connect a 10-μF capacitor between pin #6 and pin #2 of the op-amp.

CHANGE THE SENSITIVITY RANGES

The sensitivity of the Light Meter is determined by the resistors used in the circuit. If the three ranges used in the core project do not suit you, change resistors. The "Supplemental Information" section explains how to determine the correct resistor to use.

CONTROLLING OUTPUTS BASED ON LIGHT LEVEL

You may want to design a device that controls a lamp, heater, or fan, based on the light level being received. Project 11 of *How to Build a Better Mousetrap* explains how to control DC electrical devices from the computer game port. Project 11 of this book explains how to control stepper motors from the game port. A simple example of the kind of projects possible would be a system that moves a shade over a greenhouse when the illuminance is too high.

CHANGE THE PHOTODIODE SO YOU CAN DETECT IR AND UV LIGHT

The photodiode recommended for this project is designed to be most sensitive to light in the visible range. Most photodiodes are sensitive to both visible and infrared light. If you use the same circuit with one of these diodes, you will have a sensor that approximates a *radiometric* light sensor; that is, one that measures the total light energy received. There are photodiodes that are designed to respond only to infrared light. There are even UV-sensitive photodiodes that might be used for tanning alarms or other devices.

OTHER PHOTOSENSITIVE DEVICES

In the "Background Information" section we mentioned other devices that can be used for light intensity studies, such as photoresistors, phototubes, and phototransistors. Some manufacturers and distributors of these products are the following: Hamamatsu Corporation, EG&G VACTEC, Silonex, TRW, Mouser, Clairex, Motorola, All Electronics, and Radio Shack. Try building light sensors that use these other devices. Refer to data sheets or call the company to make sure that you get the best photosensitive device for your needs. The data sheets describing the sensors usually include suggested circuits.

Program Improvements

The LIGHT program on the *Chaos in the Laboratory Programs #2* disk is a fairly simple program that can be improved by adding new features. The "Program Design Notes" section below explains how the program operates. It should help you in making modifications. Some suggested changes are:

◊ Make the program beep when the reading is off scale. This will remind the user to change ranges.
◊ Allow the illuminance data to be stored in memory and then saved on a disk and reloaded later.
◊ Time how long it takes between readings and add a variable to keep track of elapsed time.
◊ If your computer has a "clock card" to keep track of real time, change the program so that it reads the clock time and stores it along with the illuminance reading.
◊ Program the computer to record the maximum and minimum illuminance readings encountered.
◊ Add a mode to the program that graphs the illuminance readings versus time.
◊ Add a mode to the program that displays the illuminance in large low-resolution graphics block digits so they can be read at a distance. Use the QUIZ program as an example of how to produce the large characters.
◊ Change the program so that it acts like a photographic light meter; that is, if you enter a shutter speed and film speed, it calculates the proper f-stop setting.

Additional Experiments

MAKING MEASUREMENTS WITH THE LIGHT METER IN SUNLIGHT

Using this meter in the high range (0 to 150,000 lux), there are many different project ideas that can be pursued. You can measure the illuminance during the day or on specific dates during various seasons to watch the solar energy levels change. You can also measure the absorption levels of different cloud types as they filter the sunlight to the ground. Another possibility is measuring the light

level changes during solar or lunar eclipses. You might even use the Light Meter as an alarm, warning when too much solar radiation has been received during sunbathing.[6]

LINEARITY TESTS

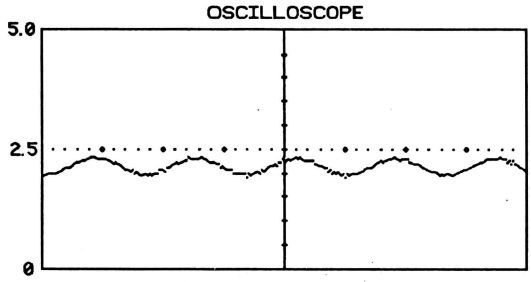
This project assumes that the voltage out of this circuit is proportional to the illuminance at the light sensor. Is it really? One way to test this would be to set up four identical lamps clustered together. Position the Light Sensor to read the illuminance produced by them. There should be no other light sources around. Measure the light level caused by each lamp individually. Then make the same measurements with 2, 3, and 4 lamps on. Graph or compare the results numerically to check the linearity of the Light Sensor.

THE INVERSE SQUARE LAW

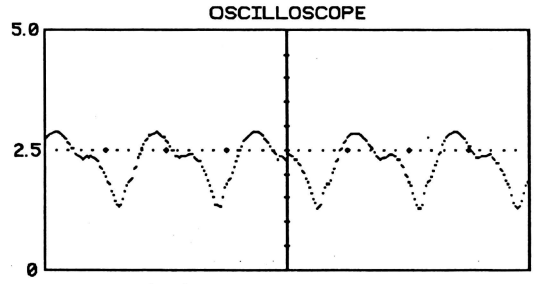
The inverse square law says that the intensity of energy radiated from a point source is inversely proportional to the distance squared. It is frequently discussed in physics. We can test the inverse square law very easily using the Light Meter. Use a darkened room as your environment, and a simple 8-volt lamp (such as Radio Shack #272-1136) or other similar lamp as your light source. Set your Light Meter on the low range (0 to 600 lumens). By varying the distance from the photodiode to the source, you can measure the illuminance as a function of distance. Graph this on graph paper, or use a computer graphing program, to plot the results.

STUDYING INDOOR LIGHT SOURCES

Another topic to study using the Light Meter is indoor light sources. If these light sources use AC line voltage, they are probably not steady light sources. The line voltage varies like a sine wave 60 times a second, so the lights actually change brightness very quickly. Graphs of the illuminance from both a fluorescent source and a tungsten source are shown below. These were made using this project, an A-to-D Converter, and the SCOPE program from the *Chaos in the Laboratory Programs #2* disk.



Light from a Tungsten Lamp



Light from a Fluorescent Lamp

Some topics for investigation:

- What causes the particular shape of the wave?
- What is the period of these fluctuations?
- Why does the illuminance never go completely to zero?
- What causes the complex shape of the fluorescent lamp graph?

USING THE LIGHT METER IN PHOTOGRAPHY

Use the Light Meter in a photographic darkroom. When you are using an enlarger, you often have to do test exposures to determine how long the enlarger lamp should be left on in exposing the photographic paper. It would be interesting to experiment with this Light Meter as a way of estimating the exposure times. The LIGHT program could even be modified so that it calculates and displays the

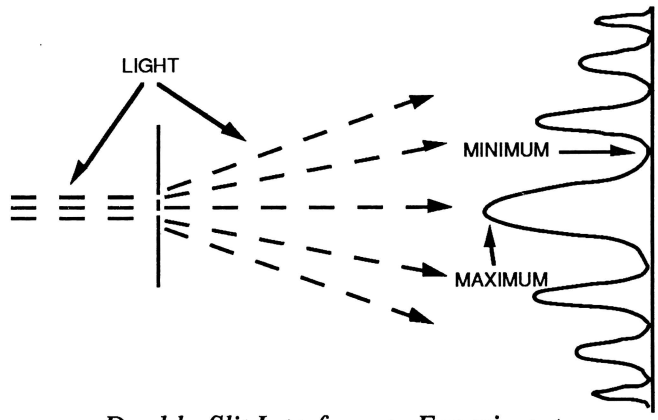
[6]The Light Sensor core project is most sensitive to visible light and does not really measure the intensity of ultraviolet light. Is it reasonable to assume that the intensity of ultraviolet radiation in sunlight varies in proportion to the illumination?

correct exposure time. It could be programmed to do the timing and beep when it was time to start and stop the exposure.

The Light Meter could also be used for determining the shutter speed for a camera.

LIGHT INTERFERENCE AND DIFFRACTION PATTERNS

This project can also be used to map out light interference and diffraction patterns. If you are doing a two-slit interference experiment, you can move the photodiode along the screen where the light projects from the two slits. You should detect a series of alternating bright and dark regions. It would be helpful to write a program to graph the data.



Double-Slit Interference Experiment

You can go one step further in this experiment by mounting the photosensor so that it moves under the control of a stepper motor (see Project 11). Use the stepper motor to gradually and steadily move the sensor across the interference pattern to produce an accurate graph of light intensity vs. position.

STUDYING LASER LIGHT

Most inexpensive laboratory lasers use tubes of Helium-Neon. The light emitted by these lasers is red, and the Light Meter can be used to measure its intensity. Diffraction and interference experiments can most easily be done using laser light.

Some lasers have inputs for modulation of the beam. This could lead to some interesting experiments; for example, a modulated laser beam can carry a sound wave that is picked up by the Light Meter and turned back into sound.

MALUS'S LAW FOR POLARIZING FILTERS

Verifying the *Law of Malus* for polarizing filters is another interesting experiment you can perform. The Law of Malus describes how the intensity of light transmitted through two polarizing filters varies. Investigate how the light transmitted through two polarizing filters changes as you rotate one of the filters relative to the other. Compare the experimental results with the predictions using the Law of Malus. Refer to college physics texts for more information.

STUDY PHOTOCHROMIC SUNGLASSES

Many popular sunglasses change their density as you move from bright sun to indoors. Use the Light Meter to study this photochemical reaction. How fast does the change occur? By what fraction is the transmittance changed? Is the reaction temperature dependent? Which wavelengths of light cause the change?[7]

[7]There is an article on photochromic sunglasses in *The Physics Teacher*, v. 28, April 1990, p. 200: "Sunglasses that Switch On and Off" by John Harris and Vittorio Zanetti.

Supplemental Information

More about the Light Meter Circuit

When analyzing an op-amp circuit such as the one we constructed, there are two assumptions about op-amps that you can make. In a high-gain linear op-amp circuit, the two input terminals (+ and -) are always at the same voltage. When examining a circuit, we can assume that this holds true most of the time. Also, you can assume that no current flows into or out of the op-amp input terminals. In reality, a small current does indeed flow, but we can usually ignore it.

These assumptions can help explain the operation of the Light Meter circuit. The data sheets for the S1133 photodiode indicate that its current output in the "short circuit" mode is very linear over a wide range of light intensities. There is almost no voltage drop across the photodiode, because the op-amp attempts to keep its input terminals (pins #2 and #3) at the same potential, and the + terminal (pin #3) is grounded. Since the op-amp ideally draws no current through its inputs, we know all the electrons flowing through the photodiode travel through the resistor, and not into the op-amp input terminal. The photodiode really acts as a "negative current source," supplying electrons which then flow through the resistor in amounts proportional to the illuminance. The op-amp output voltage is simply the product of the photodiode current and the resistance of the feedback resistor that is switched into the circuit. The capacitor in the circuit is used to filter out any electrical noise that may be picked up on the +5V power line.

We chose the CA3140AE op-amp because of its MOSFET input circuitry which gives it extremely high input impedance. Other op-amps did not work as well in our tests.

The choice of range-setting resistors can be explained as follows: The photodiodes we recommend for this project produce a current of 0.005 μA/lux. For experiments measuring small variations in illuminance in dim light, the range we might wish to use is approximately 0 to 600 lux. 600 lux corresponds to a photodiode current of 3.0 μA (600 lux * 0.005 μA/lux). The op-amp produces a maximum signal of about 3.0 volts, so we will choose this as our maximum voltage. Therefore, the value of our feedback resistor should be about (3 V / 3.0 μA) or 1-MΩ. This is the feedback resistor used when the Light Meter is in its most sensitive range. Each volt corresponds to 200 lux. One of the other feedback resistors has a value of 100-kΩ. When this resistor is switched into the circuit, it is in parallel with the 1-MΩ resistor. This produces an equivalent resistance of $(10^6 * 10^5) / (10^6 + 10^5) = 0.909 * 10^5$ Ω.or about 90 kΩ. This gives us a range about 1/10 as sensitive: 0 to 6000 lux. This is excellent for general-purpose use, such as checking indoor illumination levels. The third resistor used is 4.02-kΩ, which gives us total feedback resistance of 4.0 kΩ in parallel with the 1-MΩ resistor. This resistor was chosen to provide the maximum signal with the brightest illuminance readings in bright sunlight (150,000 lux). The table below summarizes the three ranges:

range	feedback resistance	volts/lux	lux/volt
0 to 600	1 MΩ	$5 * 10^{-3}$	200
0 to 6000	90.9 kΩ	$4.5 * 10^{-4}$	2220
0 to 150,000	4 kΩ	$2 * 10^{-5}$	50,000

Program Design Notes

The LIGHT program will do the calculations necessary to convert the voltage readings from the Light Meter to illuminance measurements in lux. Here is a section-by-section explanation of the program.

Lines 10 to 90: Introductory menu
This section asks which voltage input device you are using. It sets the variable VU$ to "A" if you are using the A-to-D Converter, and "V" for a Voltage Monitor or VIU.

Lines 100 to 990: Initialization
This section loads the appropriate machine-language program, and clears the screen and keyboard buffer.

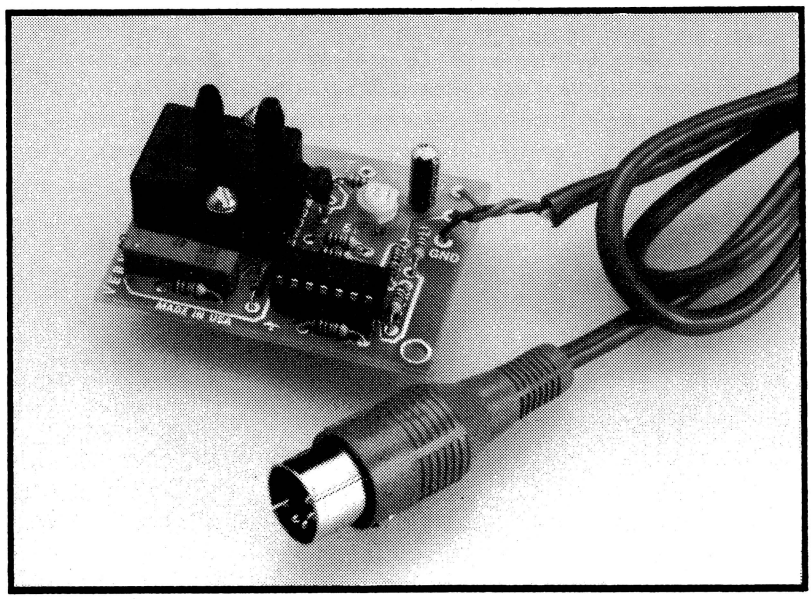
Lines 1000 to 1990: Display illuminance measurements
This section simply reads the illuminance and displays it on the monitor. It also checks for the <SPACE BAR> to be pressed. This is used to change from one range to another depending upon the position of the SPDT switch. The user can change the position of the switch, and then press the <SPACE BAR> and change the calibration of the program automatically so that it displays the proper illuminance in lux.

Lines 10000 to 10990: Voltage check subroutine
These lines are used to do the check of the voltage from the voltage input device. It also converts from volts to lux. This routine calculates the voltage slightly differently depending on the voltage input device being used.

The constant of proportionality used to convert volts to lux is determined by the feedback resistance set by the switch setting. For the low range, we use 200 lux/volt. The conversion factor for the middle range is 2220 lux per volt. Finally, the conversion factor for the high range is 50,000 lux /volt.

The following variables are used in the LIGHT program:

C	Counts from the voltage input device
CH	The voltage input device channel to read
D$	Control D, used for disk access
F	Current frequency reading from voltage input device
GS	Temporary variable used for setting normal GS speed
KB	Value of character in keyboard buffer
LUX	Value of illuminance
QQ$	Temporary string variable for displaying range
RANGE	Binary variable for range setting
V	Present voltage reading
VU$	Voltage input device - "A" = serial A-to-D, "V" = voltage monitor or VIU



Project 7

Barometer/Pressure Sensor

This project consists of building an apparatus that will allow you to measure air pressure using a computer. A commercial pressure sensor is used, which produces a small voltage that is proportional to the pressure. A two-stage instrumentation amplifier circuit with offset adjustment is used to increase the voltage signal and adjust the sensitivity of the circuit. The voltage can be fed into the Apple II using any of several voltage input devices. Depending on how you set up things, the system can be used as a pressure gauge for studying large pressure changes or as a computerized barometer. Programs are included on the *Chaos in the Laboratory Programs #2* disk which will read out barometric pressure directly. You can combine this project with temperature sensors, a light meter, and humidity sensors to make your own weather station.

Project Requirements

Voltage Input Device

You will need one of the following voltage input devices:

- A-to-D Converter from Project 5 of this book
- Voltage Monitor from Project 7 of *How to Build a Better Mousetrap*
- Vernier Software 16-pin Apple II Voltage Input Unit
- Vernier Software 9-pin Apple II Voltage Input Unit (requires a one-line program change—see *Appendix F – Teacher's Guide*)

This project can also be used with other software all of these laboratory interfaces. All of these devices accept the 5-pin DIN plug used for this project.

- Vernier Software Voltage Input Unit (Apple II or IBM-compatible)
- Vernier Software MultiPurpose Lab Interface (Apple II or IBM-compatible)
- Universal Lab Interface (Macintosh, or any computer with a RS-232 port)

Computer

This project can be done with any Apple II computer, provided you have the appropriate voltage input device.

Background Knowledge

- Electronics: If you use the Vernier Software Barometer Parts Kit, which includes a custom printed circuit board, this project is fairly easy. You simply place the components at the proper locations on the circuit board and solder the leads. Experience with soldering is recommended. If you do not use the custom circuit board, the project will be more difficult. In that case, you will need to use just the schematic diagram as your guide to assembly.

- Programming: None required for the core project; however, some of the project extensions require experience with BASIC programming.

Parts

- ☐ SenSym SCX15ANC Pressure Transducer (Available from SenSym, 1255 Reamwood Ave., Sunnyvale, CA 94089, 408-744-1500)
- ☐ TLC274 quad op-amp
- ☐ 14-pin socket (Radio Shack #276-1999)
- ☐ LM385Z-2.5 Voltage Reference Diode
- ☐ 200-kΩ, 1/4 watt, metal-film resistor (red, black, black, orange, brown)
- ☐ (7) 100-kΩ, 1/4 watt, metal-film resistor (brown, black, black, orange, brown)
- ☐ 2.2-kΩ, 1/4 watt, metal-film resistor (red, red, black, brown, brown)
- ☐ 470-Ω, 1/4 watt, metal-film resistor (yellow, violet, black, black, brown)
- ☐ 1-kΩ, 1/4 watt, carbon resistor (brown, black, red, gold)
- ☐ 5-kΩ potentiometer
- ☐ 100-kΩ multi-turn potentiometer
- ☐ 10-μF capacitor
- ☐ (2) bolts and nuts for connecting pressure sensor to the circuit board
- ☐ circuit board (This project is greatly simplified if you use the custom circuit board that is included in the Vernier Software Barometer Parts Kit. If you use a general-purpose circuit board, make sure it is large enough for all of the components.)
- ☐ cable with 5-pin DIN plug for connection to voltage input device

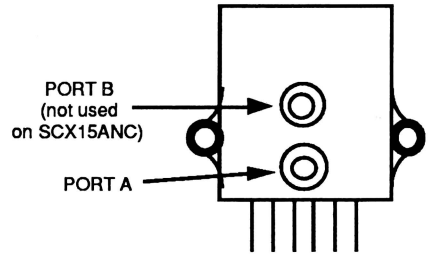
Note: A Barometer Parts Kit containing all of the components needed for this project, including a custom-printed circuit board, is available from Vernier Software. This kit sells for $42.00 (Order Code CBR-DIN). An order form is provided at the back of this book.

Tools and Supplies

Low-power soldering iron, solder, wire cutters, wire strippers, screwdriver, and pliers or an alligator clip to act as a heat sink. A volt-ohm meter may be useful if troubleshooting is required. Clear plastic tubing with an inside diameter of approximately 3/16 inch (0.47 cm) is useful for many experiments.

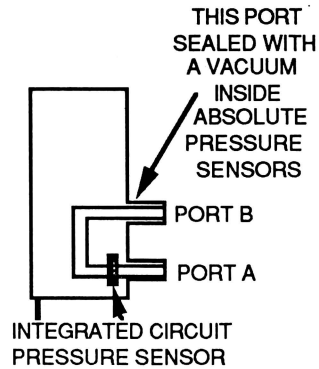
Background Information

The heart of this circuit is the SenSym SCX pressure sensor. These pressure sensors are designed to produce a voltage that varies linearly with pressure. They include special circuitry to minimize errors caused by changes in temperature. The SCX series of pressure sensors are available covering pressure ranges from 0-1 to 0-100 pounds per square inch. SCX pressure gages are also available for either absolute pressure measurement or differential pressure measurement. Each SCX pressure sensor has two ports on the back.



SCX Pressure Sensor

If the sensor is designed for differential pressure measurement, both of the ports are connected to the internal sensor and the sensor responds to the difference in the two pressures. If the sensor is designed to measure absolute pressure (as in this project) port B is sealed off with a vacuum inside.



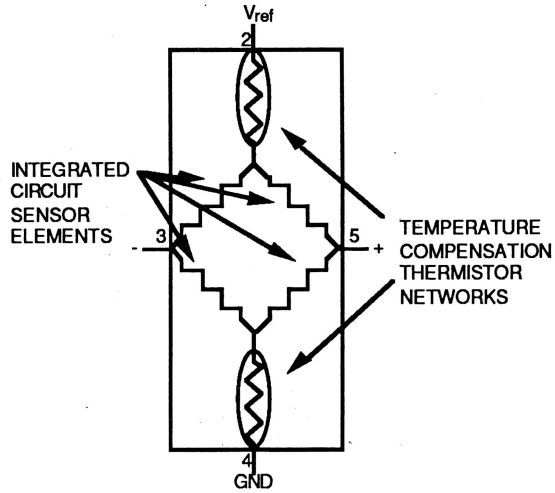
SCX Absolute Pressure Sensor

Pressure Units

Pressure can be measured in many different units. Since this project is designed for use primarily as a computerized barometer, we are going to use the units in which pressure is most often reported in weather reports. In the United States, those units are inches of mercury. This unit of air pressure is based on the traditional liquid mercury barometer—it is the pressure required to cause a mercury column to rise one inch. For comparison, conversion factors to a number of other pressure units are listed below:

1 PSI (pound per square inch)	= 2.036 inches of mercury (in. of Hg)
1 atmosphere	= 29.92 in. of Hg
1 millibar	= 0.02953 in. of Hg
Pascal (= N/meter^2)	= 295.3 in. of Hg
K Pascal	= 0.2953 in. of Hg
mm of Hg	= 3.937 x 10^{-2} in. of Hg

How does the pressure sensor produce a voltage output? The diagram indicating what is going on electrically inside the sensor is shown below:



The Internal Electrical Connections of the Pressure Sensor

A stable voltage is applied between pins #2 and #4 of the sensor. The heart of the sensor is an integrated circuit. This chip contains the equivalent of four resistors. It is mounted in the sensor so that when a pressure is applied to the sensor, the chip flexes (very slightly) and the resistance of these four resistors change. This change causes the voltages at pins #3 and #5 to change. We will be using the voltage changes across pins #3 and #5 as our measure of pressure.[1] Two temperature compensation networks, made up of *thermistors* (temperature sensitive resistors) are connected to the top and bottom of the pressure sensor element. The thermistor networks have been specifically designed to correct for any change in the output of the sensor caused by a change in temperature.[2]

The specific sensor we will use for this project is the SCX15ANC. The "A" in the part number indicates that it measures absolute pressure. The "15" designates that the sensor's range is 0 – 15 psi (0 – 30.54 in. of Hg). Normal atmospheric pressure is about 14.7 psi (29.9 in. of Hg), so this unit will work nicely as a barometer. The specifications for the SCX15ANC are given below:

- Pressure range 0 to 15 PSI or 0 to 30.54 inches of mercury
- Maximum pressure that the unit can tolerate without permanent damage: 30 psi or 61 in. of Hg
- Sensitivity (using a 2.5-volt reference voltage): 1.25 millivolts/psi or 0.614 mv/inch of mercury
- Long term stability: +/– 0.1% full scale reading
- Response time: 100 microseconds

To read the pressure via computer, all we have to do is build an amplifier that will step up the voltage out of the sensor to the range that the voltage input device can measure accurately. We will build an *instrumentation amplifier* for this job. An instrumentation amplifier is designed to provide high-gain amplification of the voltage difference between its two input signals. At the same time, it should be insensitive to other factors, produce very little "noise," and be very stable. Our instrumentation amplifier will be designed around a readily available and inexpensive integrated circuit—a quad op-amp. All of the voltage input devices that can be used to read the signal from the barometer can monitor voltages in the range of 0 – 3 volts. Our amplifier has been designed to produce an output signal that fits that range.

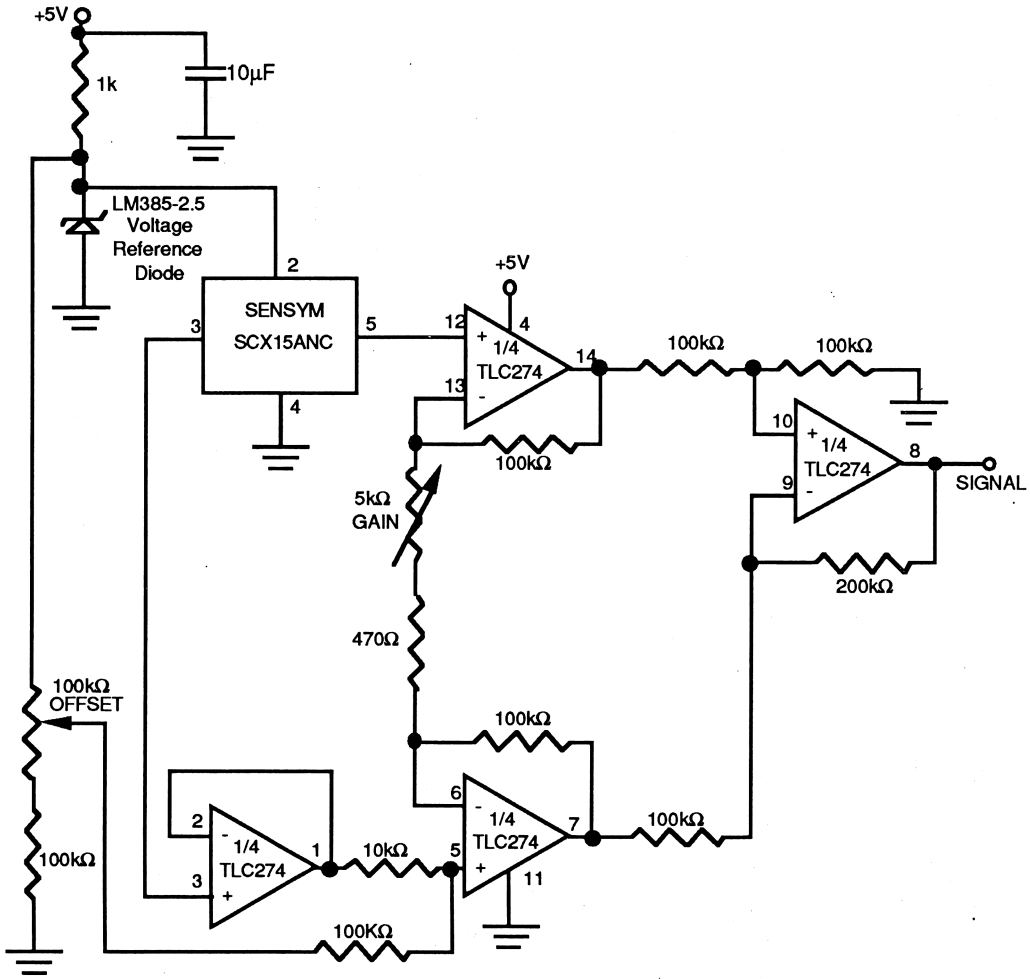
The complete schematic diagram for the Barometer/Pressure Sensor circuit is shown below. The upper left corner of the diagram shows the circuit that supplies the stable 2.50 volts. The four op-amps shown in the diagram are all contained in one integrated circuit chip—the TLC274. Three of the op-amps are used to make an instrumentation amplifier. The gain of this amplifier is determined by the

---

[1]The pressure sensor is similar to a strain gage bridge circuit. The four resistances of the circuit are arranged in a Wheatstone bridge circuit. This type of circuit is also used frequently in instrumentation; for example, see Project 9 of *How to Build a Better Mousetrap.*

[2]One of the project extensions is to use these thermistor networks as a temperature measurement device.

setting of the 5-kΩ potentiometer. It determines how much voltage change is caused by a given change in pressure. The gain can vary from 82 to 853 times. A 100-kΩ potentiometer is used as an offset control. It can be adjusted to increase or decrease the voltage out of the circuit. It is normally used just to set the initial output to match a known pressure reading.[3]



Schematic Diagram of the Barometer Circuit

There are two ways you may want to use this circuit. To use this project as a barometer for studying weather, the circuit must be set for its maximum sensitivity. Atmospheric pressure does not change very much. Air pressure rarely goes below 27.5 inches of mercury or above 31.5 inches of mercury. This is a change of only about 13%. Except in times of violent storms, the change per day is usually less than two percent. This kind of measurement, especially made over a long time period, requires a good instrument. The SCX15ANC has been designed to be very consistent in its response. It includes internal features to minimize changes in response caused by temperature variations. The amplifier circuit with the gain potentiometer turned all the way up has a gain of 853. That means that any voltage change produced by the pressure sensor will be multiplied by 853 times. As mentioned above, the SCX15ANC produces an output change of 0.614 mV per inch of mercury, so:

overall output = 0.614 mV/ in. Hg * 853 = 524 mV/in. of Hg

That is, the output voltage will vary about a half volt per inch of Hg atmospheric pressure change. We will adjust the offset potentiometer to make sure that the voltage output range of the circuit falls within the proper range.

---

[3]The "Supplemental Information" section of this project has more information on the amplifier circuit.

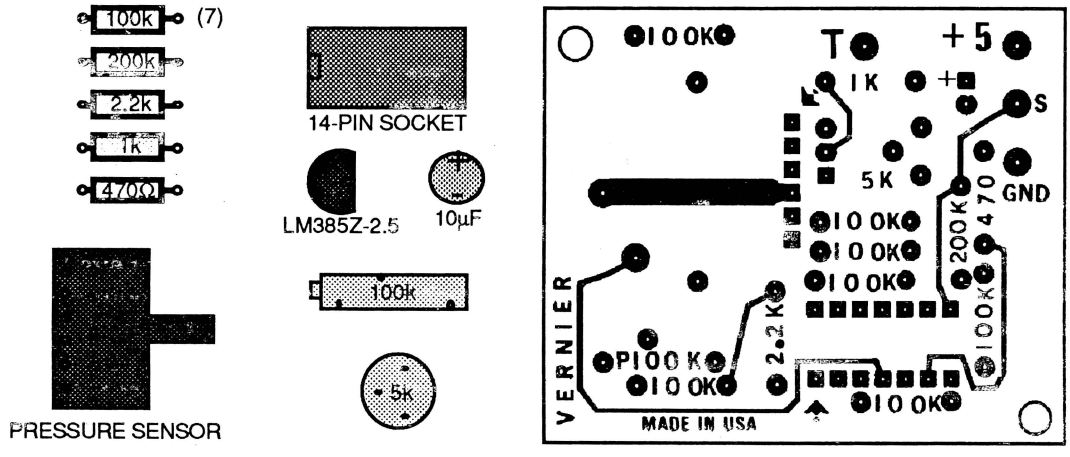
There is a program on the *Chaos in the Laboratory Programs #2* disk named PRESSURE that is designed for use with the circuit set up for maximum sensitivity. To calibrate the circuit to use this program, all you have to do is run the program and adjust the offset potentiometer so that the reading displayed on the computer monitor matches the known pressure (obtained from another barometer, or a weather report).

If you want to use this project in a situation where the pressure varies more than about 4 inches of Hg, you need to turn down the gain of the amplifier. Examples include Boyle's and Charles' law experiments, and vacuum experiments. When you turn down the gain, you probably will also need to adjust the offset potentiometer to keep the voltage in the 0 – 3.5 volt range. When the gain is turned down, you should use only the voltage displayed by the PRESSURE program. The voltage displayed will vary in a linear way as the pressure changes. You would have to change one line of the PRESSURE program to make it read the correct pressure for your gain setting.

The SCX15ANC pressure sensor is fairy durable, but it is designed only for use with noncorrosive gases such as air, helium, nitrogen, etc. Do not get it wet and do not apply too much pressure to it. Note that it is designed only to measure pressures from approximately standard atmospheric pressure to a complete vacuum. If you want to measure pressures higher than this, you should use a different (less sensitive) pressure sensor. Sensors with different pressure ranges are available from SenSym and other manufacturers.

Step-by-Step Instructions

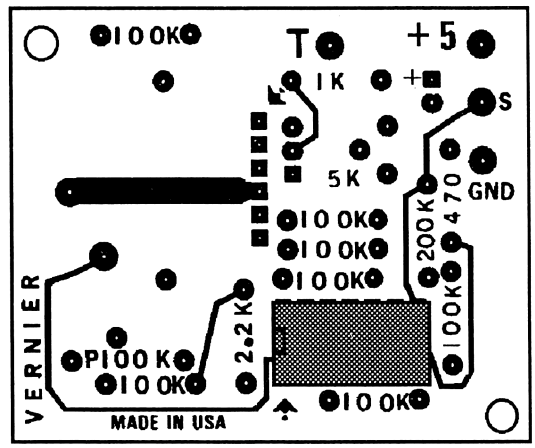
Overview: These instructions assume you are using a Vernier Software Barometer Parts Kit, which includes a custom circuit board. If you are building the project using a general-purpose circuit board or a prototyping board, you should use the schematic diagram in the "Background Information" as a guide.



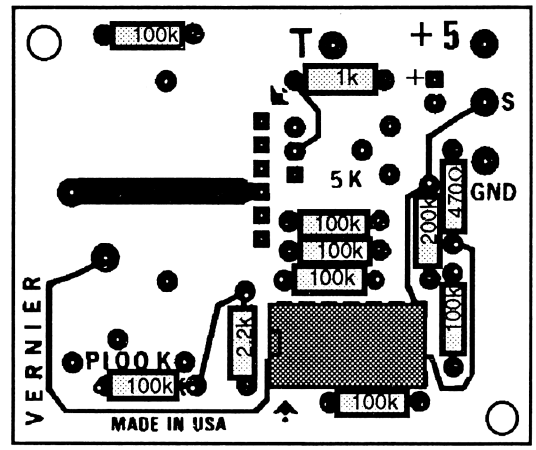
The Barometer Circuit Board - Component Side and Parts to be Added

☐ 1. **Identify the parts.** The Vernier Software Barometer Parts Kit contains all of the components needed for this project. Check the parts list included with the kit and identify each part.

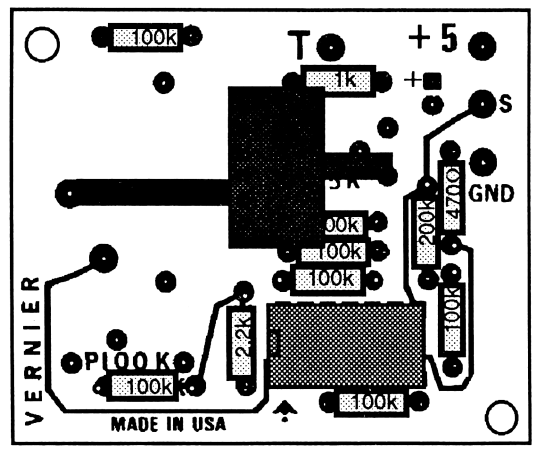
☐ 2. **Solder the socket to the board.** Position the 14-pin socket in the position shown below. The socket and all of the other components go on the side of the board that has lettering. Note that the notch in the socket should be near the center of the circuit board. Solder all 14 pins to the other side of the circuit board.



☐ 3. **Add the resistors to the circuit board.** There are 11 resistors in this circuit. The circuit board has the value for each resistor labeled on the component side. Use these labels and the diagram below to position each resistor on the circuit board. Push the leads into the holes as far as possible. Solder all the leads. Cut off the excess leads as close to the circuit board as possible.

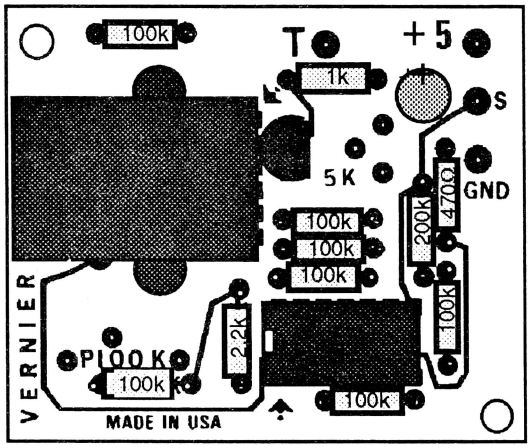


☐ 4. **Add the pressure sensor to the circuit board:** The SCX15ANC pressure sensor has 6 pins in a line. The pins are numbered on the case of the sensor. Position it on the component side of the circuit board in the position shown below. Pin 1 goes at the top of the circuit board. Solder all 6 pins to the solder side of the circuit board.

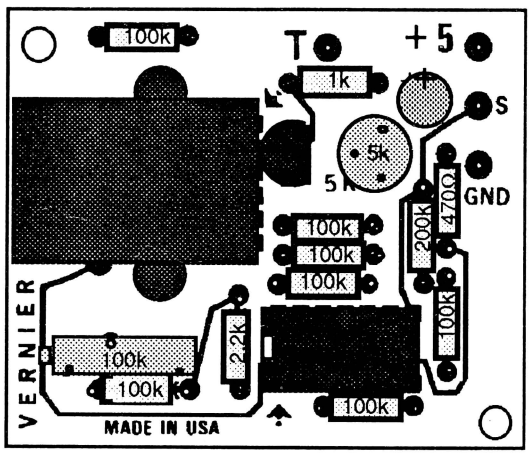


☐ 5. **Bolt the pressure sensor to the circuit board.** Bend the pressure sensor leads so that the flat side of the body of the sensor lays down against the circuit board. Bolt the pressure sensor to the circuit board using the two bolts through the body of the sensor and the circuit board.

☐ 6. **Add the voltage reference and capacitor.** Note that the flattened side of the LM385Z-2.5 voltage reference should be oriented to the right. Use a heat sink when soldering it. The capacitor has a "–" mark on it. That side goes toward the bottom of the circuit board. Solder these two components in place and cut off the excess leads of these components as close to the circuit board as possible.

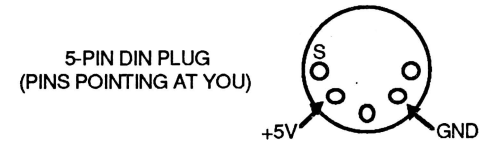


☐ 7. **Add the two potentiometers.** Position the two potentiometers as shown below. Note that the 5-k potentiometer is a small, round potentiometer and the 100-k potentiometer is rectangular. Position the 100-k, rectangular potentiometer with the adjustment screw near the edge of the circuit board. Positioning the 5-k, round potentiometer can be confusing. If you orient it correctly it will slip into the three holes with almost no bending of the leads. To double-check its orientation, look carefully at the base of the potentiometer. There should be a small notch on the right side of the 5-kΩ potentiometer. Solder both potentiometers in place.



☐ 8. **Connect the DIN cable.** This circuit connects to the voltage input device using a 3-conductor cable with a DIN connector on the end. The three connections to use on the cord are shown below. In Vernier Software Parts Kits, the following color code is used to identify these three leads: black = GND, red = signal, orange = +5V.

If you are using your own cable, use an ohmmeter to identify which wire on your cable connects to each terminal on the plug.



Solder the three connections to the locations marked "+5", "S", and "GND" on the right side of the circuit board.

☐ 9. **Plug in the IC.** Carefully place the TLC274 IC into the socket. Be sure to orient pin #1 at the proper corner of the socket, marked with a notch (near the center of the circuit board).

☐ 10. **Test the completed circuit.** Turn off the computer. Plug the barometer circuit into the voltage input device. Turn on the computer and run the PRESSURE program on the *Chaos in the Laboratory Programs #2* disk to read the voltage. If you are using the A-to-D Converter (Project 5 of this book), select A at the initial menu. If you are using a Voltage Monitor, select V. Select D - DISPLAY PRESSURE from the menu offered. The computer should begin displaying a voltage and a pressure. For now, use the voltage values and ignore the pressure values. Adjust the gain potentiometer (5 kΩ) to near its middle position. The computer should display a voltage reading between 0 and 4 volts. Adjust the offset potentiometer (100 kΩ) until the computer reads about 1 volt at normal air pressure. This is a multi-turn potentiometer, so you may have to make several complete rotations of the screw to reach the right setting. Blow gently on the input port nearest the pins on the pressure sensor. The voltage should increase slightly. Suck on the input port and the voltage should decrease.

☐ 11. **Set up the circuit for use as a barometer.** To use the circuit as a barometer to study the small changes in atmospheric pressure, adjust the round 5-kΩ gain potentiometer to it lowest resistance by rotating the adjustment as far counterclockwise as it will go. This makes the circuit most sensitive to pressure changes. You will then need to adjust the offset potentiometer (the rectangular one) to keep the voltage within the range of about 1 to 3 volts. You can check the sensitivity of the unit by gently blowing into the input port. If the voltage does not increase, you have the gain set too high and you should reduce it.

With the gain potentiometer adjusted for maximum sensitivity, the gain of the circuit is 853; therefore, the voltage out of the circuit should change by 0.614 mV/inch Hg * 853 = 0.524 V/ in Hg. The program PRESSURE on the disk assumes that the circuit is set up in this way. To adjust the circuit so that it will work with the PRESSURE program and read out pressure directly, follow the procedure below. For greatest accuracy, you should let the barometer circuit warm up for about 10 minutes before doing this adjustment.

a) Run the PRESSURE program.
b) Make sure the gain potentiometer (the round one) is set for maximum sensitivity (fully counterclockwise).
c) Determine the local air pressure at your location. If you have another barometer available, you can use it. Otherwise, check the local TV or radio weather report, or call a nearby airport.
d) Adjust the offset potentiometer (the rectangular one) until the reading displayed by the barometer program matches the known pressure.

This setting should provide a fairly accurate pressure reading. If you want to do a more involved calibration, refer to the "Project Extensions" below.

☐ 12. **(optional) Set up the circuit for use as a general-purpose pressure gauge.** If you do not plan to use the circuit for weather studies, experiment with the gain and offset adjustments. Turning the 5-kΩ potentiometer clockwise will make the circuit less sensitive to pressure changes. Whenever you adjust the gain potentiometer (5 kΩ), you will need to adjust the offset potentiometer (100 kΩ) to keep the voltage within the range of about 1 to 3 volts. It is possible to measure the whole range of pressures from 0 to slightly more than normal air pressure.

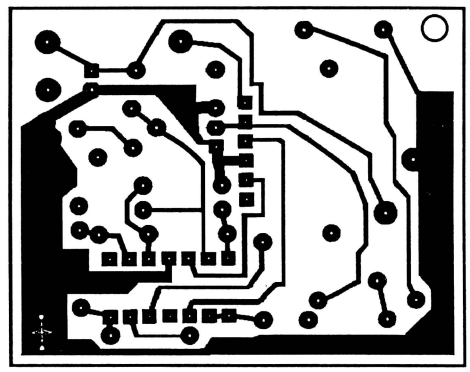
Troubleshooting

The following tests are recommended to help you spot problems. All voltages are relative to ground (GND) unless otherwise stated. All voltage measurements should be made with the system connected and the computer on.

If your system always reads a constant voltage (or zero), check the following:

◊ Is your IC inserted properly? Pin #1 is marked with a dot or indentation. Is it near the center of the circuit board?
◊ Make sure you have all the resistors in the right places. Check the color codes listed on the sheet in the parts kit list or in *Appendix B* of this book. The resistor values are labeled on the circuit board.

◊ Visually double-check all of the solder joints on the circuit. There are a lot of connections and it is easy to miss one in your soldering. Also, check for solder bridges between points on the circuit board that should not be connected. The figure here shows the back of the circuit board as it was before you started soldering. Compare it to your circuit board to check for solder bridges.



The Back of the Circuit Board

If none of these problems are found, it is time to check the voltages at various points on the circuit. The negative lead of the voltmeter should be connected to the GND line from the voltage input device.

Position	Voltage Should Be	Comments
IC pin 4	+5 V	This is the power supply lead on the DIN cable.
IC pin 11	0 V	This is the ground lead to the IC.
IC pin 3 or 12	approximately 1.25 V	These are the signals from pins #3 and #5 of the pressure sensor.
IC pin 5	1.25 to 2.5 V (offset voltage)	Check the solder joints of the 100-kΩ potentiometer and the 100-kΩ resistor at the top left of the circuit board. This voltage should vary as you turn the 100-kΩ potentiometer.
IC pin 8	0 to 3 V	This is the output voltage from the IC. It should read the same voltage that the program displays. It should change as you change the pressure or as you adjust the potentiometers. If it does not, and all of the checks above are ok, perhaps the IC is inserted incorrectly or is damaged.
Middle pin of LM385-2.5	2.5 V	This is the reference voltage. It should be very close to 2.5 volts. If it is not, check the orientation of the LM385-2.5. The flat side should be away from the pressure sensor.
Pins 3 and 5 of the SCX15ANC	1.25 V	These are the signals out of the pressure sensor. They are normally about half of the reference voltage (i.e., 1.25 volts). If not, check the orientation of the pressure sensor and its solder joints.

Project Extensions

Circuit Modifications

INCREASING THE SENSITIVITY OF THE BAROMETER

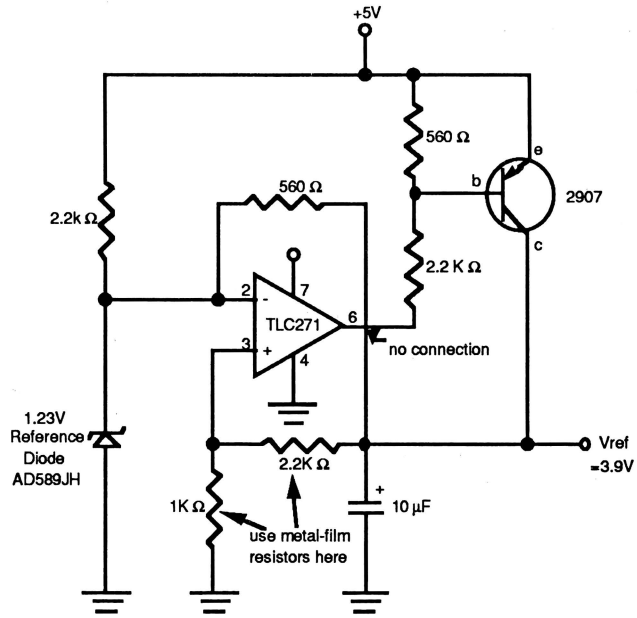
If you replace the 470-Ω resistor in the barometer with a slightly smaller value resistor, the gain of the circuit will be increased. Be sure to use a metal-film resistor for best temperature stability. Note that as the gain of the circuit increases, electrical noise will also increase, so at some point the extra gain may not be an improvement. If you make a change in the gain of the circuit you should also change the PRESSURE program so that the program reads pressures correctly.

USE THE TEMPERATURE OUTPUT OF THE SCX15ANC

Pin #1 of the SCX15ANC pressure transducer is connected at the end of one of the temperature compensation thermistor networks inside the sensor. Its voltage will vary in a nearly linear way as the temperature of the sensor changes. The output voltage ranges from about 1.88 volts at 0°C to 1.92 at 25°C. Calibration for each individual sensor would be a requirement. You probably would want to build an amplifier into the circuit to make the voltage change larger.

AN IMPROVED REFERENCE VOLTAGE CIRCUIT

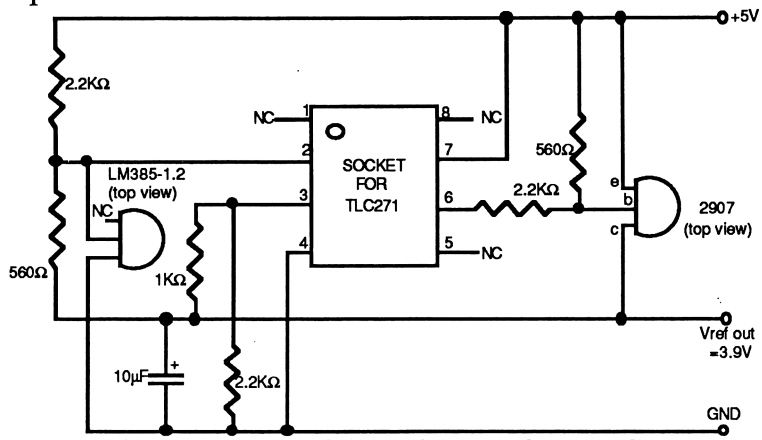
The core Barometer/Pressure Sensor project uses a LM385Z-2.5 voltage reference diode to provide a steady voltage to the pressure sensor. The circuit below is an alternative voltage reference circuit that has two advantages over the simpler voltage reference circuit: (1) It supplies a reference voltage of 3.9 volts instead of 2.5 volts. The higher reference voltage increases the output signal from the pressure sensor. (2) It can supply a lot more current while still holding the voltage steady. This circuit can supply a steady 3.9 volts even with the current of 60 mA. The pressure sensor used in the core project draws only about 1 mA, so the ability to supply a lot of current is not essential for this project, but may be for other similar circuits.[4]



Schematic Diagram of the Voltage Reference Circuit

[4]For example, the Strain Gage circuit described in *How to Build A Better Mousetrap* can be improved by using this circuit to supply the excitation voltage to the strain gage bridge. The strain gage bridge in that project draws about 50 mA of current.

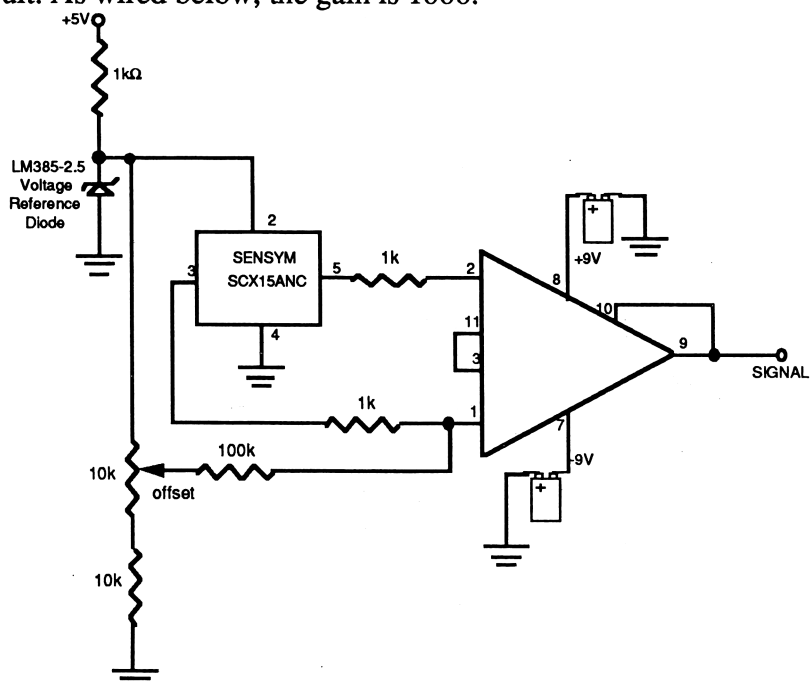
A working diagram of the voltage reference circuit is shown below. The working diagram is a top, component-side view of the circuit. Note carefully the orientation of the transistor, the capacitor, and the precision voltage reference. It is essential that metal-film resistors be used for the two resistors connected to pin #3 of the op-amp. Changing the reference voltage will require a change in the programs used to read pressure.



Working Diagram of the Voltage Reference Circuit

USE A COMMERCIAL INSTRUMENTATION AMPLIFIER

There are several integrated circuit manufacturers who make ready-to-use instrumentation amplifiers. One example of such an amplifier is the Analog Devices AD524.[5] It is essentially an improved, assembled version of the instrumentation amplifier that we describe in the "Step-by-Step Instructions." It has *programmable gain*, which means you select the amplification factor by the way you connect one of its pins. The circuit below shows the total circuit needed for this project using an AD524. Notice that the circuit is much simpler than the core project circuit, since the whole instrumentation amplifier is built into the one IC. Note that the AD524, like most commercial instrumentation amplifiers, requires a dual power supply. This means you need to use two 9-volt batteries in the circuit. As wired below, the gain is 1000.



Commercial Instrumentation Amplifier Circuit

[5]The AD524 is manufactured and sold by Analog Devices Inc., Two Technology Way, P.O. Box 280, Norwood, MA 02062. They can also be purchased from Vernier Software for $15 each.

OTHER PRESSURE SENSORS

Many other pressure sensors can be used in place of the SCX15ANC used in the core project. SenSym makes SCX pressure sensors with ranges from 0 – 1 psi to 0 – 100 psi. The SCX30ANC is one that may be very useful. It measures absolute pressure up to 30 psi (61 in. of Hg). SCX sensors that measure the difference in pressure between the two ports are also available. All of the SCX sensors have the same pin configuration, so you can use any of the sensors on the same circuit board. The SenSym address and phone number are listed in the parts list at the beginning of this project.

Other companies that sell pressure transducers that are similar include: Motorola (MPX series of pressure sensors), Nova Sensors (PH series of sensors), Omega Engineering, Micro Switch, ICSensors, Endevco, and Foxboro/ICT.

Program Improvements

The PRESSURE program on the *Chaos in the Laboratory Programs #2* disk is a fairly simple program that can be improved by adding new features. The "Program Design Notes" section below explains how the PRESSURE program operates. It should help you in making modifications. Some suggested changes are given below:

◊ Allow the pressure data to be saved on a disk and reloaded later.
◊ If your computer has a "clock card" to keep track of real time, change the program so that it reads the clock time and stores it along with the pressure reading.
◊ Program the computer to record the maximum and minimum pressure readings encountered.
◊ Add a mode to the program which graphs the pressure readings versus time. You should use high-resolution screen #1 (HGR) to avoid memory conflicts.
◊ Add a mode to the program that displays the pressure in large low-resolution graphics block digits so they can be read at a distance. Use the QUIZ program on the *Chaos in the Laboratory Programs #1* disk as a model.

Additional Experiments

BOYLE'S LAW

Boyle's law is a classic physics/chemistry concept that can be demonstrated using the Barometer/Pressure Sensor. The law states that if the temperature remains constant, the pressure of a container of a gas is inversely proportional to its volume. Set up a sealed container of air such that you can change its volume. Take data on how the pressure changes with volume change. One easy way to do this is to use a large hypodermic syringe as the air container. These syringes have the additional advantage that they are often marked in volume units (cc).

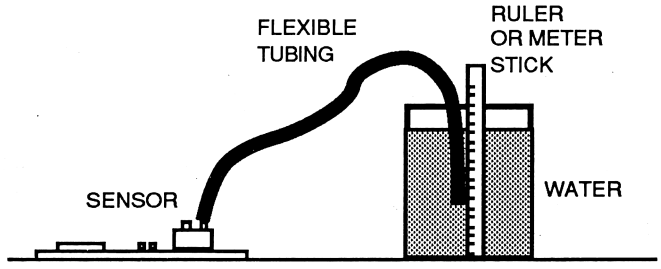
CHARLES' LAW

Charles' law states that if the volume remains constant, the pressure of a container of a gas is directly proportional to its absolute temperature. Set up a sealed container of air such that you can change its temperature. Take data on how the pressure changes with temperature changes. Remember that all temperatures should be measured using the Kelvin temperature scale.[6] The same large hypodermic syringe which work well for Boyle's law experiments can be used with Charles' law. Just place the whole hypodermic syringe in a warm or cold water bath. You may need to temporarily glue the syringe so that its volume does not change.

[6]Temperature in kelvins = Celsius temperature + 273.15.

CALIBRATION USING A TUBE IN WATER

A more accurate calibration of your barometer can be made by using the apparatus diagrammed below:



When the tube is connected to the input port of the Pressure Sensor and the end of the tube is put underwater, the pressure reading will increase. The pressure increase will be 0.02896 inches of mercury for every centimeter below the surface of the water. Test this with your circuit and the PRESSURE program. Write down the normal air pressure. Now place the end of the tube underwater and note how far the end of the tube is below the surface (in centimeters). Write down the pressure reading reported by the program. Calculate the following ratio:

$$\text{ratio} = \frac{\text{Pressure change as measured by the PRESSURE program}}{\text{Pressure change as calculated from depth} * 0.2896 \text{ V/in Hg}}$$

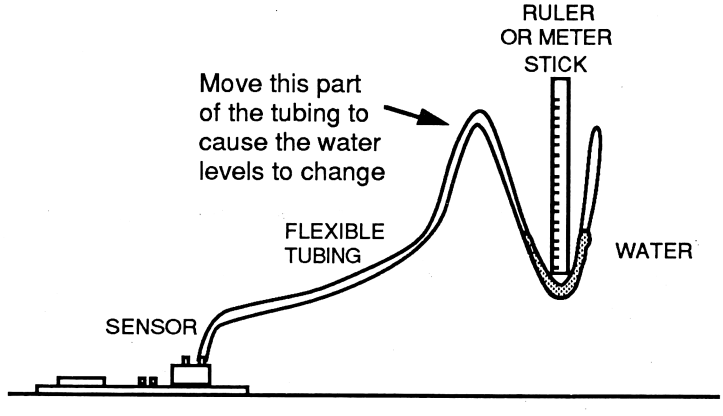
Line number 190 of the PRESSURE program can be changed to better calibrate your barometer. This line originally reads:

```
190 RATIO = 1.00 : REM CORRECTION FACTOR
```

If you want to correct the program so that it works with your barometer circuit to read pressures more accurately, change the value 1.00 in this line to the ratio you calculated above. List the program to make sure that you have made the change correctly, and then save the program on a disk.

CALIBRATION USING A J TUBE

Another way to check the calibration of your barometer circuit is to use a J tube as shown below:



The tube is open on the end. When the tube is connected to the input port of the Pressure Sensor, the left and right water levels will both be the same. If the tubing connecting the J-tube apparatus is moved, the two water levels will change due to the change in the pressure in the air column on the right side. Be careful to keep the water from traveling down the tube into the pressure sensor. The difference in the heights of two water levels is a measure of the pressure. The pressure change is equal to the weight of a column of water with a unit area of the same height. Note that it is the difference in pressure that you always measure. Each centimeter corresponds to 0.02896 in. of Hg.

PRESSURE IN LIQUIDS

The idea of measuring the pressure in a liquid was used as a calibration method in the section "Calibration using a Tube in Water" above. This same idea can be expanded to measure pressure change in other liquids. In general, the pressure change should be:

pressure = density * depth * acceleration due to gravity

FREE FALL STUDIES

How does the acceleration of a falling object change when you remove the air around it? The Barometer/Pressure Sensor, combined with accurate timing equipment, can help you study the acceleration of objects in air. Set up a tube that you can evacuate. The object should fall inside the tube. Use this project to measure the pressure and determine the acceleration at different pressures. Use a very light object for the greatest change in acceleration. Photogates can be used to determine the acceleration.

BAROMETRIC TIDES

The atmosphere, like the oceans, goes through tidal cycles. Can you detect changes in the barometric pressure that are regular and tied to tidal effects? For this project you will want to have the computer plot pressure vs. time graphs.

Supplemental Information

Additional Information on the Amplifier Circuit

The instrumentation amplifier circuit used in this project is based on a fairly standard design found in most electronics textbooks. This amplifier has two stages. The gain of the first section is given by the equation $gain_1 = 1 + 2 * (R / R_g)$, where R_g is the resistance between pins #6 and #13 of the IC, and R is the resistance of the feedback resistors for the first stage. We use 100-kΩ (±1%) feedback resistors for this circuit.

The second stage has a gain of just two (from the ratio of the 200-k to the 100-k resistors), so the overall gain of the amplifier is:

$$\text{overall gain} = (1 + 2 * (100\ k\Omega / R_g)) * 2$$

Note that the instrumentation amplifier cannot produce a voltage above about 3.5 volts, so turning the gain up too high can limit the range of pressures that can be measured. Because of this, an offset voltage is added to the barometer circuit. This offset voltage is fed into pin #3 of the IC, along with the buffered signal from the pressure sensor. Increasing the offset voltage pulls the final output voltage lower.

When you are using the circuit as a barometer, we recommend that you set the gain potentiometer at its lowest resistance, which should be zero. If you look at the schematic diagram of the circuit, you will see that there is a 470-Ω resistor in series with the gain potentiometer. That means that R_g is 470 ohms when the gain potentiometer is set to zero. A metal-film, ±1% resistor is recommended for this 470-Ω resistor, so its resistance should be quite accurate. The overall gain of the circuit with the gain potentiometer set this way is then:

$$\begin{aligned}\text{overall gain} &= (1 + 2 * (100\ k / 470)) * 2 \\ &= 853\end{aligned}$$

The sensitivity of the SCX15ANC (using a 2.5-volt reference voltage) is 0.61 mv/ in. of Hg, according to the SenSym data sheets. This yields an overall sensitivity for the sensor and the amplifier of 0.614 mV/in. Hg * 853 = 524 mV/in. Hg. This is the value assumed in the PRESSURE program.

Minor correction of this value can be done by changing one line of the program as explained in the calibration project extension.

If you want to use this project as a general-purpose pressure sensor that covers the whole range of the SCX15ANC, the gain must be turned way down. In that case, you want the final output voltage to vary from 0 to about 3 volts as the pressure goes from 0 to 30 inches of Hg.

full scale range of the SCX15ANC = 0.614 mV/in. of Hg * 30.5 in. of Hg = 18.7 mV = 0.0187 V

desired gain = 3 volts / 0.0187 volts = 160

Using this value with the overall gain equation and solving for R_g yields R_g = 2.4 k. You can easily set the gain for this value, since a 5-kΩ gain potentiometer is used in this circuit.

Program Design Notes

The PRESSURE program on the disk will do the calculations necessary to convert the voltage readings from the barometer/amplifier to pressure measurements. Here is a section-by-section explanation of the program:

Lines 10 to 90: Introductory menu
This section asks which voltage input device you are using. It sets the variable VU$ to "A" if you are using the A-to-D Converter, and "V" for a Voltage Monitor. It also sets LOMEM to just above the first graphics screen. This sets the start of the variable storage memory.

Lines 100 to 990: Initialization
This section loads the appropriate machine-language program used to read the voltage. It also initializes variables and displays the main menu. Line number 130 sets the number of readings to store in memory. This is now set at 2000. The array begins in memory just above the first graphics screen. If you want to make major additions to the program, or use the second graphics screen, you will have to reduce the size of this array and possibly set LOMEM above the second graphics screen.

Lines 1000 to 1990: Display Pressure
This section simply reads the pressure and displays it on the monitor.

Lines 2000 to 2990: Display Pressure and Save in Memory
This section displays the pressure readings and also saves them in memory. Up to 2000 pressure readings can be saved in memory. When this mode is selected, you will be asked:

```
ENTER DELAY TIME (TIME BETWEEN PRESSURE
READINGS STORED IN MEMORY) [1]
```

This option will allow you to save pressure measurements made over a long time period. This is useful in weather studies and for checking the stability of the barometer. If you enter a time delay of 5, for example, the reading of the barometer will be checked continuously, but the readings will be saved in memory only every 5 seconds. Usually, you do not need this delay. If you simply press <RETURN> or enter a 1, the readings will be saved in memory once a second.

After you enter this value, the program will begin monitoring the pressure. Any time the pressure reading is saved in memory, the line on the data table is marked <- SAVED. The screen will display both the time (in seconds) and the pressure. The times measured by the program are only approximate.

Lines 4000 to 4990: Table of Data
This section lets you display a table of data collected using the S = SAVE PRESSURE DATA IN MEMORY option. Both the pressure and the time at which the pressure was measured are displayed.

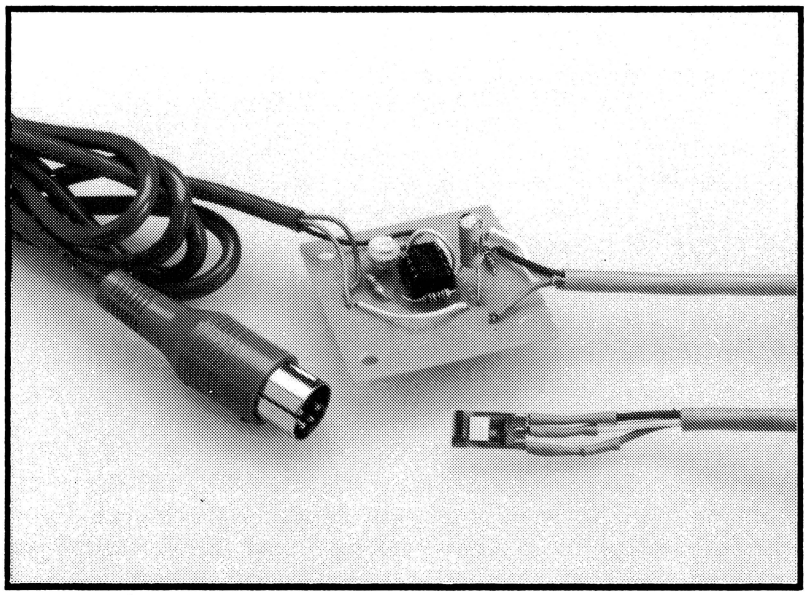
Lines 10000 to 10990: Voltage Check Subroutine
These lines are used to do the check of the voltage from the voltage input device. The voltage is calculated slightly differently depending on which voltage input device you are using.

Throughout these programs, the output voltage of the barometer is assumed to be linear with respect to pressure. In fact, these programs assume that the overall sensitivity of the pressure sensor and amplifier circuit combination is 0.524 V/inch of Hg. We have also set up the program to expect the barometer circuit offset potentiometer to be set so that 25.0 inches of Hg corresponds to 0 volts. The equation used in converting voltage to pressure is then:

pressure = 25.0 + voltage / 0.524

The following variables are used in the PRESSURE program:

C	Counts from the voltage input device
CH	Channel read using a voltage monitor
D$	Control D, used for disk access
F	Current pressure reading
GS	Used for setting the GS speed
I	Delay loop counter
IN	The pressure reading
J	Data line counter
M$	Mode of operation
N	Number of data lines
NR	The number of readings to store
Q$	Dummy input string
RT	The multiplication ratio
T	Time (in seconds)
T( )	Time data saved in memory
TD	Time delay for saving data
V	Present voltage reading
V( )	Pressure data saved in memory
VU$	The voltage input device



Project 8

Magnetic Field Sensor

In this project you will build a magnetic field sensor. It uses a linear output Hall effect transducer that produces a small voltage proportional to the magnet field. This voltage is then amplified with a simple op-amp circuit to a level that can be monitored by the computer. The basic sensor can be used to explore the field around permanent magnets or electrical devices. One project extension involves increasing the amplification of the circuit, making it possible to directly measure the earth's magnetic field.

Project Requirements

Voltage Input Device

You will need one of the following voltage input devices:

- A-to-D Converter from Project 5 of this book
- Voltage Monitor from Project 7 of *How to Build a Better Mousetrap*
- Vernier Software 16-pin Apple II Voltage Input Unit
- Vernier Software 9-pin Apple II Voltage Input Unit (requires a one-line program change—see *Appendix F – Teacher's Guide*)

This project can also be used with other software and any of the following laboratory interfaces. All of these devices accept the 5-pin DIN plug used for this project.

- Vernier Software Voltage Input Unit (Apple II or IBM-compatible)
- Vernier Software MultiPurpose Lab Interface (Apple II or IBM-compatible)
- Universal Lab Interface (Macintosh, or any computer with a RS-232 port)

Computer

This project can be done using any Apple II computer, providing the appropriate voltage input device is used.

Background Knowledge

- Electronics: This is a relatively easy project, but it does have quite a few parts. Experience with soldering components to a circuit board is recommended.
- Programming: None required for the core project. A simple BASIC program change is required if you want to change the sensitivity of the system. Some of the other project extensions require experience with BASIC programming.

Parts

- ☐ Hall effect transducer (Micro Switch SS94A1 Linear Hall Effect Transducer)
- ☐ TLC271 op-amp (This is a low-power CMOS op-amp made by Texas Instruments. It is specifically designed to be used with a single +5-volt power supply.
- ☐ 8-pin IC socket (Radio Shack #276-1995)
- ☐ circuit board (Any small circuit board designed for use with DIP ICs can be used. The step-by-step instructions assume you are using one-half of a Radio Shack #276-159.)
- ☐ cable with 5-pin DIN connector (one-half of a DIN patch cord [Radio Shack #42-2151] or a 5-pin DIN plug [Radio Shack #274-003] with 3 wires soldered onto it)
- ☐ 1-kΩ, 1/4 watt metal-film resistor (brown, black, black, brown, brown)
- ☐ 10-kΩ, 1/4 watt metal-film resistor (brown, black, black, red, brown)
- ☐ (2) 47-kΩ, 1/4 watt metal-film resistor (yellow, violet, black, red, brown)
- ☐ 10-kΩ potentiometer, single-turn, PC board mount (Digi-Key #36C14)
- ☐ 10-μF electrolytic capacitor (Radio Shack #272-1025)
- ☐ 1 meter of 3-conductor cable, preferably shielded
- ☐ (3) short pieces of wire for connections on the circuit board
- ☐ heat shrink tubing (5 cm) or electrical tape

Notes:

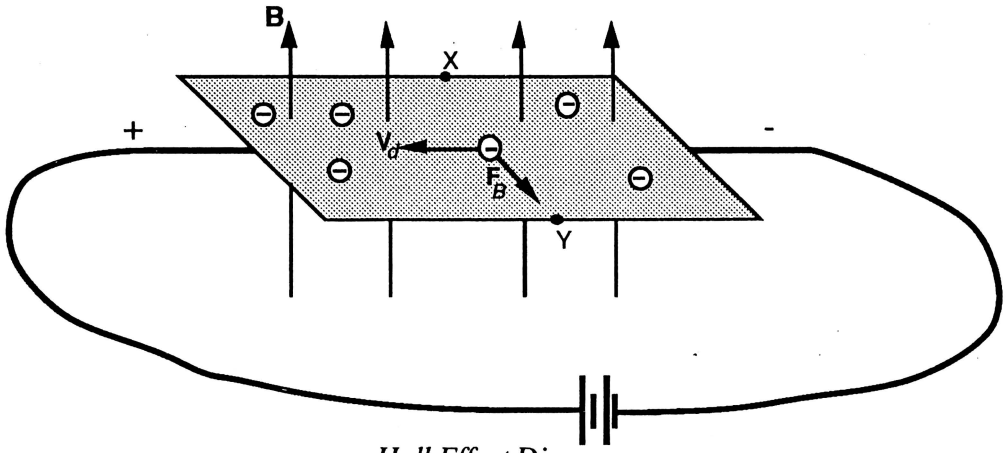
- A Magnetic Field Sensor Parts Kit containing all of the components needed for this project is available from Vernier Software. This kit sells for $25.00 (Order Code CMG-DIN). An order form is provided at the back of this book.

Tools and Supplies

Low-power soldering iron or soldering pencil, solder, wire cutters, wire strippers, ohmmeter, and magnets for experimentation.

Background Information

The Hall effect was discovered in 1897 by Edwin Hall, a 24-year old graduate student. He found that when a very thin rectangle of gold through which current was flowing was placed in a magnetic field, an electric potential difference appeared between the two opposite edges. This is because the magnetic field causes a force on moving electrons. Just as a magnetic field diverts an electron beam, the electrons moving in the conducting sheet are disturbed. A measurable potential difference results from this interaction. This is called the Hall effect.



Hall Effect Diagram

In the diagram above, a metal sheet is connected in a circuit with an external power supply. Electrons drift toward the left in the sheet, as indicated by vector $\mathbf{V}_d$. A magnetic field **B** is perpendicular to the sheet. The magnetic field causes a force shown as $\mathbf{F}_B$ to be exerted on the electrons. This causes an uneven distribution of electrons in the sheet. Electrons build up on the side of the sheet labeled Y, leaving an excess of positive charges on the side marked **X**. This results in a potential difference between points X and Y. If a sensitive voltmeter is connected between X and Y, a small electric potential that is proportional to the magnetic field can be measured.

If the magnetic field is reversed, the electrons migrate in the opposite direction, resulting in a reversed potential difference. The sensor is affected only by the component of the magnetic field that is perpendicular to the metal sheet.

The Hall effect was an important discovery in theoretical physics, because it provided a way to tell whether the charge carriers moving in a metal were negative or positive. If the moving charge carriers in a metal were positive, the Hall effect voltage would have been the opposite polarity.[1]

The Hall effect had few practical applications outside physics for more than 70 years. In the 1950s and 1960s, applications started to appear in the form of instruments to measure magnetic fields, position sensors, switches, and even keyboards that use the Hall effect. The SS94A1 Hall effect transducer that we will be using consists of a Hall effect sensor such as the one described above, a built-in amplifier, and output transistors. The voltage produced by the Hall effect transducer is proportional to the intensity of the magnetic field, but it is too small to be read directly by the computer. For this transducer the output is 0.0031 volts/gauss. The Hall effect sensor is in a small ceramic rectangle on the transducer. The transducer always measures the component of the magnet field perpendicular to this rectangular object. If you are familiar with the concept of "magnetic lines of force," you can think of this as measuring how many of these lines cut through the rectangular Hall effect sensor.

This project consists of combining the Hall Effect transducer with an external amplifier to step up the voltage so that it can be easily monitored by the computer. The voltage varies in a linear way with magnetic flux density. We will adjust the circuit so that it produces 2.0 volts when there is no magnetic field. The voltage will then vary up or down depending on the magnitude and direction and of the magnetic field.

---

[1]If you have learned the left-hand rule or right-hand rule for the force on a moving charge caused by a magnetic field, try to verify this statement by referring to the Hall effect diagram. For more information on the Hall effect and its importance in physics, refer to a college physics text such as *Fundamentals of Physics*, Halliday and Resnick, 3rd Edition, Wiley and Sons, 1988 or *Physics for Scientists & Engineers*, Raymond A. Serway, Saunders College Publishing, 1990.

For the core project, the sensitivity of the amplifier is set so that it is useful for studying the fields near common permanent and electromagnets. The amplifier gain is 10, and the output of the Magnetic Field Sensor is 0.031 volts/gauss or 3.1 x 10^{-6} volts/tesla. The range of fields that can be measured is ±64 gauss (±6.4 3 x 10^{-3} tesla). The program MAGNET on the *Chaos in the Laboratory Programs #2* disk converts these voltages to gauss and tesla and displays them on the monitor.

Magnetic Field Units

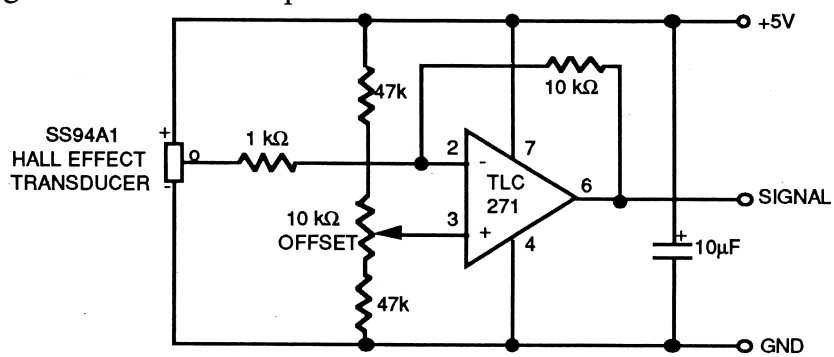
A number of different units are used when measuring magnetic fields. The specific quantity that this sensor measures is referred to as *magnetic flux density*.[2] The SI metric unit for magnetic flux density is the *tesla*. An older cgs metric unit, *gauss*, is still commonly used. One gauss = 10^{-4} tesla. One reason that the unit gauss is still popular is probably that the magnetic field near the earth's surface is roughly one gauss. We will use both units in this chapter and in the MAGNET program. Another unit that is used extensively by geophysicists is the *gamma*. One gamma = 10^{-5} gauss or 10^{-9} tesla. Geophysicists report variations in the earth's magnetic field in gamma. They are typically looking for minor changes that may indicate the characteristics of the ground below.

Some typical values for magnetic flux densities are given below:

- At the surface of the earth: 0.55 gauss (5.5 x 10^{-5} T) is typical; the range is 0.3 to 0.7 gauss
- 3 cm from a small refrigerator magnet: 5 gauss (5 x 10^{-4} T)
- 3 cm from a neodymium "super magnet": 40 gauss (4 x 10^{-3} T)

One of the project extensions is to increase the gain of the external amplifier to make the Magnetic Field Sensor more sensitive. The gain is increased from 10 to 200. With this amplification, the output of the Magnetic Field Sensor is 0.625 volts/gauss, and the range of fields that can be measured is ±3.2 gauss. Increasing the amplification in this way causes two complications. The first is that the background noise is also amplified. Greater amplification increases the noise as well as the signal. The second complication can also be a useful feature. The earth's magnetic field can be measured. An amplification of 200 results in almost one volt change as the sensor is moved in the earth's magnetic field. You can then use it to determine magnetic north and the magnetic dip in your area. On the other hand, if you want to measure anything other than the earth's field, you will have to keep the sensor in one position as measurements are made. Otherwise, the magnetic field of the earth will interfere with your readings.

The complete Magnetic Field Sensor schematic diagram is shown below. The amplification factor is specified by the ratio of two fixed resistors (1 kΩ and 10 kΩ in this diagram). The voltage from the Hall effect transducer when no magnetic field is present is roughly 2.5 volts—halfway between +5V and ground. A magnetic field can either raise or lower this voltage, depending on the direction of the field. Since we want to amplify only the voltage changes from this starting voltage, an offset voltage is applied to pin #3 on the op-amp. The 10-kΩ potentiometer is used to adjust the voltage at pin #3 so that it is the same as the voltage at pin #2 when no magnetic field is present. The op-amp amplifies the difference in voltage between these two pins.



Magnetic Field Sensor Schematic

[2]This quantity is sometimes called *magnetic induction*.

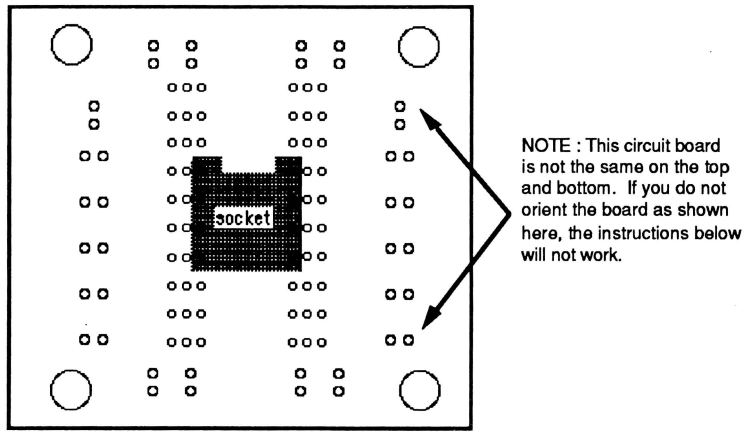
There are a number of interesting experiments that can be done using the Magnetic Field Sensor. You can explore the fields around permanent magnets, electromagnets, TVs, motors, and speakers. You can do an investigation into how the magnetic field inside a solenoid varies with current, number of turns, etc. You can do a quantitative study of how the magnetic field drops off as you move away from a magnet. If you increase the amplification of the circuit, you can determine the direction and magnitude of the earth's magnetic field.

Step-by-Step Instructions

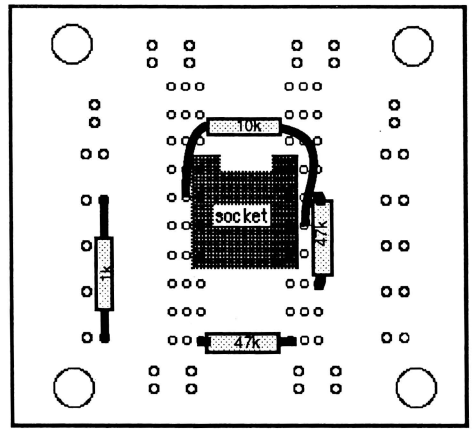
These instructions assume that you are using one-half of a Radio Shack #276-159 circuit board. If you are using a different circuit board, the layout of parts will have to be changed.

☐ 1. **Check all components.** Check the parts list given at the beginning of this project, and make sure each resistor and capacitor is the proper value.

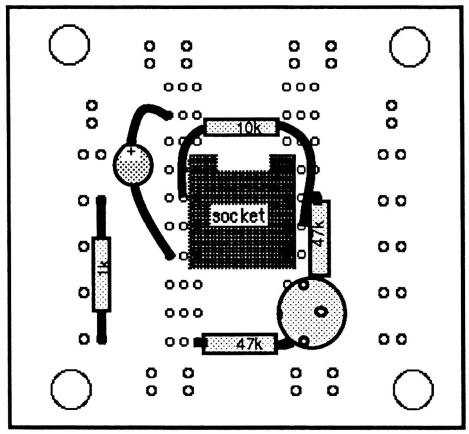
☐ 2. **Place the 8-pin socket on the circuit board.** Hold the circuit board (one-half of a Radio Shack #276-159) with the plain side up (no copper plating or writing). Place the socket on the circuit board, on the side facing you, in the position shown in the diagram below. Be certain to orient the board as shown. The top of the socket is marked with a notch to indicate pin #1. The pins are numbered counter-clockwise from pin 1 (on the upper left corner) around to pin 8. Solder all 8 of the pins to the plated side of the circuit board. Do not place the op-amp IC into the socket at this time.



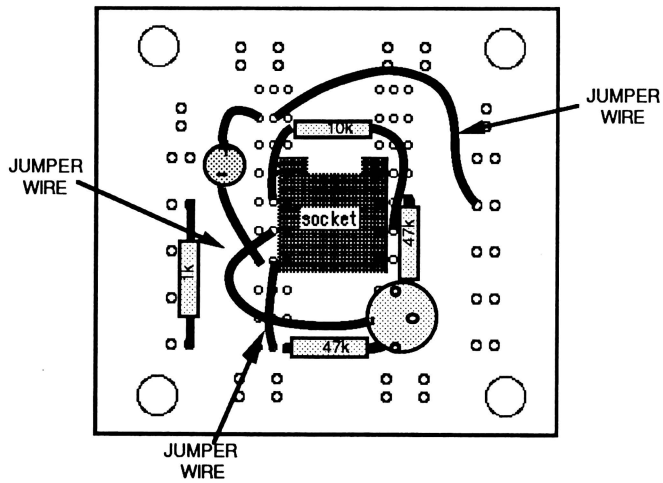
☐ 3. **Solder the four resistors to the board.** A diagram showing the placement of these resistors is shown below. You can bend the leads on the plated side so that the components stay in place while you solder them. Solder the resistors to the plated side of the circuit board. Cut off the excess leads.



☐ 4. **Solder the capacitor and potentiometer to the board.** Both of these components need to be oriented in the proper direction. A diagram showing their placement is shown below. The electrolytic capacitor has a "–" on one side. Be sure to position the electrolytic capacitor with the "–" side downward and connected to pin #4 of the socket. The potentiometer is the small, white component. Insert the three leads on the base into the three holes as shown in the diagram below. The holes into which the potentiometer should be inserted are shown by the black dots in the diagram of the potentiometer body below. To double-check its orientation, look for a small notch in the side of the potentiometer. It should be toward the left.



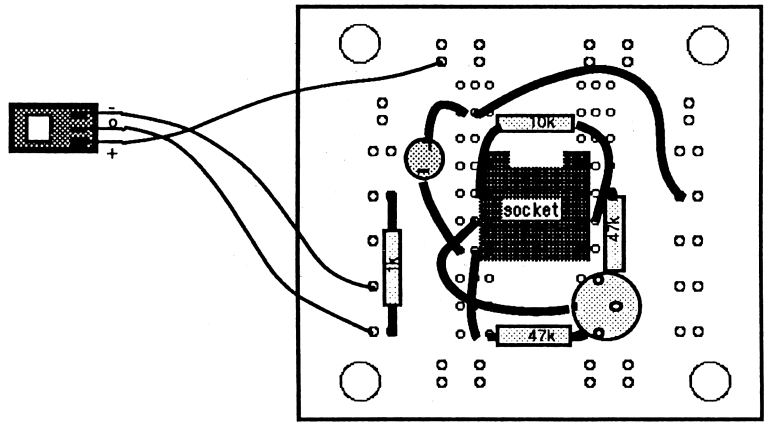
☐ 5. **Connect the jumper wires.** Three jumper wires are needed. If you prefer, you can cut the wires to just the right length to make the connections neat.[3] Solder these wires and cut off all excess wire on the solder-side of the circuit board as close to the solder joint as possible.



☐ 6. **Connect the Hall effect transducer.** Use a 1-meter long, 3-conductor shielded cable to connect the Hall effect transducer to the circuit board. Separate the three leads for several centimeters on each end of the cable. Strip about a half centimeter of insulation from each of the leads on both ends. Solder the three pins on the Hall effect transducer to these three wires. Slip heat-shrink tubing on each wire lead, so that it can later be used to insulate the connections. Move it back out of the way during soldering. To prevent damage to the transducer during soldering, it is a good idea to use a heat sink on each lead as it is soldered. A pair of pliers or a metal clip between the transducer and the soldering iron will do the job. Note the colors of the wires connecting the "+", "–", and "o" pins. The "o" pin is the signal output from the sensor. Move the heat-shrink tubing in place over the solder joint

[3]In addition to making your project more tidy, this will also help minimize electrical noise that the wires may pick up by acting as antennas.

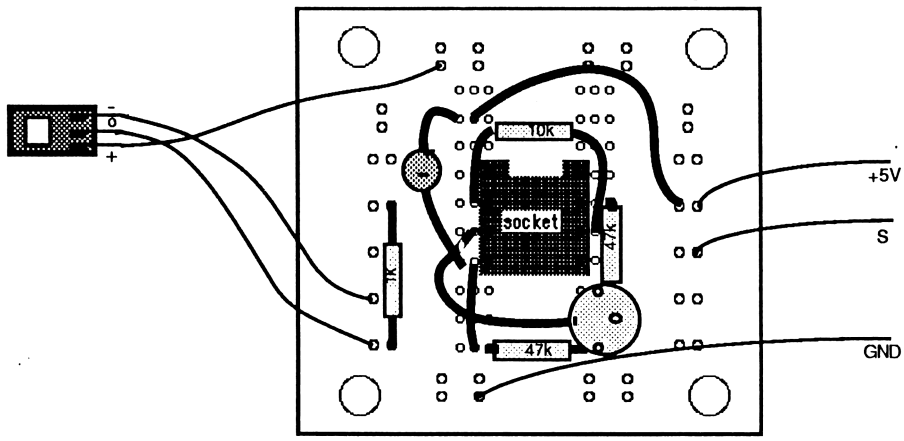
and shrink it down by rubbing it with the soldering iron. Solder the other end of each of the three wires to the circuit board as shown in the diagram below.



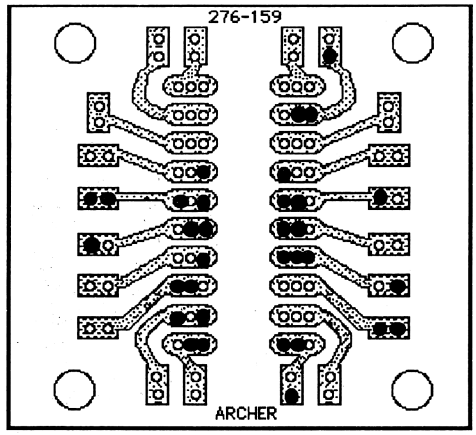
☐ 7. **Connect the DIN cable.** Connect the 3-conductor cable which will extend from the circuit to the voltage input device. The three connections to use on the cord are shown below. In Vernier Software Parts Kits the following color code is used to identify these three leads: black = GND, red = signal, orange = +5V.

If you are using your own cable, use an ohmmeter to identify which wire on your cable connects to each terminal on the plug.

Solder the three connections to the circuit board as indicated in the diagram below:



☐ 8. **Trim the leads.** Cut the leads on the plated side of the circuit and check all connections. Check for solder bridges between places that should not be connected. The diagram below shows how the back of the circuit board should look. Compare the back of your circuit board with this diagram to help in spotting wiring errors.



Completed Circuit Board as Viewed from the Back

☐ 9. **Plug in the IC.** Carefully insert the integrated circuit into the socket. Be sure to orient pin #1 at the proper corner of the socket (to the left of the notch). Make sure all eight pins go into the socket. Be careful not to bend the pins too much—they can easily break.

☐ 10. **Adjust the offset potentiometer.** Turn off the computer. Plug the Magnetic Field Sensor into the voltage input device and connect the voltage input device to the computer. Place the *Chaos in the Laboratory Programs #2* disk in the disk drive and turn on the computer. Run the program MAGNET. If you are using the A-to-D Converter (Project 5 of this book), enter "A" when asked what kind of voltage input device you are using. If you are using a Voltage Monitor, enter "V". The computer should begin displaying the voltage and magnetic flux density. The Magnetic Field Sensor is calibrated, but you need to set the potentiometer so that the program reads zero field when no magnets are nearby. Adjust the 10-k offset potentiometer until the magnetic field reads near zero when no magnetic field is present. The magnetic field strength displayed by the program will be negative if the voltage is below 2 volts, and positive if it is above 2 volts. Once the potentiometer is set properly, the output from the sensor should change as a magnet is brought near the sensor. It will increase or decrease, depending on the direction of the magnetic field. The Magnetic Field Sensor has a range of ±64 gauss ($\pm 6.4 \times 10^{-3}$ T).

☐ 11. **Use the system to study the fields around various magnets.** Experiment with different magnets and electromagnets. Notice how the field strength changes as the magnet is brought near the sensor and rotated while nearby, etc. Remember that the value reported can increase or decrease depending on the direction of the magnetic field. The small ceramic rectangle is the Hall effect sensor. The magnetic flux through that rectangle is what is measured. Can you detect the magnetic field of the earth with this system? Many ideas for experiments are in the "Project Extensions" section of this chapter.

Troubleshooting

If your circuit does not behave as you expect it to, the following tests are recommended to help you spot problems:

◊ Is your IC inserted properly? Pin #1 is marked with a dot. Is it where it should be?
◊ Have you bent one or more pins on the IC by not putting it into the socket properly?
◊ Have you adjusted the potentiometer? See step 10 above.
◊ Does your op-amp get hot when you turn things on? If it is hot, you must have a serious wiring problem, and you may even need to replace the chip after you fix the wiring error.
◊ Do you have the Hall effect sensor wired correctly? Make sure you have the +, –, and o (signal) outputs connected to the correct holes on the circuit board.

◊ Make sure you are using the correct resistors. The parts list includes the colors to help in identifying the resistors.
◊ Visually double-check all of the connections on the circuit board. Check the back of your circuit board and compare it with the diagram in step 8. If you cannot see anything in the wrong place, check again with the ohmmeter between the IC pins and the connections to the various components. If they are directly connected, the resistance will be 0.
◊ Are there any solder bridges on the plated side of the board connecting two places on the circuit that should not be connected?
◊ Are the pins on the IC all soldered properly, and not just sticking through the board?
◊ Double-check the DIN connections using an ohmmeter.
◊ Try reheating each of the solder joints.

If none of these problems are found, it is time to check the voltages on the various pins of the op-amp IC with a voltmeter. The negative lead of the voltmeter should be connected to the GND line (pin #4 on the IC). All voltage measurements should be made with the system connected and the computer on.

Pin #	Voltage Should Be	Comments
7	+5 V	This is the power supply connection. If not correct, check the +5V and GND leads on the DIN cable.
2	near 2.5V	This is the output from the Hall Effect transducer. If not correct, check the connections to the transducer.
3	2.3 to 2.7 V	This the offset voltage. It should vary over this range as the offset potentiometer is adjusted. If it does not, then make sure the potentiometer and the 47-kΩ resistors are correctly soldered into the circuit. When calibrated, this voltage should be about the same as the sensor voltage at pin #2.
6	0 to 5 V	This is the output voltage. It should change from 0 to around 5 volts as the offset potentiometer is adjusted. If this voltage seems ok, double-check the connection to the voltage input device.

If the voltage at pin #6 does change (as it should), then the problem is probably in the voltage input device or its connections. Check the voltage input device by applying a known voltage signal.

Project Extensions

Circuit Modifications

◊ Increase the amplifier gain. This is done by changing the 10-kΩ resistor between pins #2 and #6. The gain is equal to this resistance divided by the resistance between pin #2 and the Hall effect sensor. The 10-kΩ resistor used in the core project gives a gain of 10. Using a 200-kΩ resistor for a gain of 200 produces about a 1-volt variation when moved within the earth's magnetic field. If you make this change in the circuit, you should also change line 110 of the MAGNET program so that it reads GAIN = 200. You could use a switch to select two or more different resistors, so your Magnetic Field Sensor has different ranges.

◊ Decrease the amplifier gain. By changing the 10-kΩ resistor between pins #2 and #6 to a smaller resistor, you can make the Magnetic Field Sensor less sensitive so that you can study stronger magnetic fields. The sensor itself puts out 3.125 mV/gauss, and can be used to measure fields as strong as ±400 gauss. To measure the most intense fields possible, set the gain to 1 by using a 1-kΩ resistor between pins #2 and #6. With this gain setting, the output would vary by 0.003125 volts/gauss and the range would vary from 0.75 volts at -400 gauss to 3.25 at +400 gauss. If you make this change in the circuit, you should also change line 110 of the MAGNET program so that it reads GAIN = 1.

◊ Add a capacitor to smooth out the high-frequency noise. A capacitor connected between pins #2 and #6 of the IC will act as a signal averager. It will lead to more steady voltage and magnetic flux density readings. It will, however, also slow down the response time of the circuit. Do not add a capacitor if you plan to study magnetic fields that change rapidly, such as around TVs and motors.

◊ Place the Magnetic Field Sensor circuit in an experimenter box. Select a box large enough for the circuit board. Drill holes in the box for the wire to the sensor and the cord to the voltage input device.

Program Improvements

The MAGNET program on the *Chaos in the Laboratory Programs #2* disk is fairly simple. It can be improved by adding new features. The "Program Design Notes" below explain how the MAGNET program operates. It should help you in making modifications. Some suggested changes are given below:

◊ Change the units of flux density displayed—eliminate gauss or tesla, add gamma, etc.
◊ Add an "off scale" alarm, so that the program beeps when the reading reaches the maximum or minimum possible reading.
◊ Allow the data to be saved to memory.
◊ Allow the data to be saved on a disk and reloaded later.
◊ Program the computer to record the maximum and minimum magnetic field readings encountered.
◊ Add a mode to the program that graphs the magnetic field readings versus time.
◊ Add a mode to the program that displays the magnetic field readings in large low-resolution graphics block digits so they can be read at a distance. Use the program SOUND as a model. You can even use the machine-language routine LP.1200 that is used by the SOUND program.
◊ Change the program to take a number of readings and average the values. Background noise becomes a problem with high amplifications. Averaging the values helps reduce the problem.

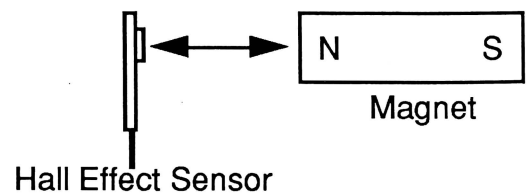
Additional Experiments

The completed Magnetic Field Sensor can be used for many interesting projects. A few examples are listed below:

◊ Explore the magnetic field intensity of electromagnets. How do they compare with the field intensities of permanent magnets? Place the sensor near an electromagnet and examine the relationship between the current and field intensity. Try using different materials for the core of the electromagnet. Using an iron core will greatly increase the field intensity.

◊ Explore the magnetic field near computer monitors, speakers, and motors. Is the field steady? If not, what is the frequency of the change in field. Investigate it using the SCOPE program. What is the shape of the field? How do they compare with the field intensities of permanent magnets?

◊ Investigate how the magnetic field of a permanent magnet changes as a function of the distance from one of the poles. Does the flux density drop off inversely with distance? Is this another example of an inverse square law in physics?



◊ Test the equations for the magnetic flux density. There are equations in most college physics texts for the flux density near a long straight wire, inside a solenoid, or inside a toroid. Look up these equations and then check them experimentally using circuits with known current passing through them.

◊ Experiment with inserting ferromagnetic materials into the magnetic field around a permanent magnet or electromagnet. When a ferromagnetic material (such as an iron bar, nail, etc.) is brought into the field, it increases the intensity. If you have modified the circuit to the high gain setting, you

can also see this effect using just the magnetic field of the earth—the field increases just by having ferromagnetic material around.

◊ Explore the magnetic field of the earth. To do this you will need to set the gain of the amplifier higher as explained in the first project extension in this chapter. Try to determine magnetic north by rotating the sensor and observing the change. The sensor measures the component of the magnetic field perpendicular to the flat surface. When you find the maximum field reading, magnetic north is perpendicular to the sensor in the direction of the green and white side. The voltage is at a minimum when the white ceramic rectangle on the sensor points toward magnetic south. Halfway between the maximum and minimum no magnetic field is measured. This occurs when the flat side of the sensor is parallel to the magnetic field lines of the earth. Note that magnetic north is probably not exactly the same direction as geographic north, and that magnetic north points somewhat downward.

What is the magnetic *declination* in your area; that is, by how many degrees does magnetic north differ from true north? How does your measurement of the declination made with the Magnetic Field Sensor compare with measurements made with a compass? How does it compare with published values for your area?

You can also determine the magnetic *dip* in your area; that is, the angle of the magnetic field lines from the horizontal. The lines of the earth's magnetic field don't point directly toward the magnetic pole—they curve toward the center of the earth. At the equator they are roughly parallel to the ground, but at the magnetic poles they point almost directly down. The deviation from horizontal is called the *paleomagnetic inclination* or "dip." The inclination in the U.S. ranges from about 60° in the south to 70° in the north.

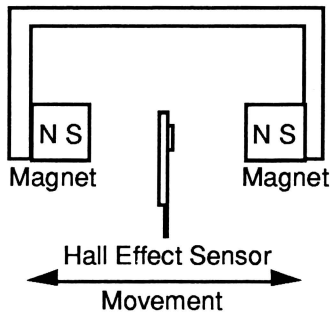
The absolute intensity of the magnetic field of the earth is a little more difficult to measure. You need to set the amplification to about 200 and calibrate the program to read magnetic field intensity. The customary unit used in geophysics is the *gamma*. After the system is calibrated, find the maximum and minimum field intensities. To find the maximum and minimum, try to visualize the magnetic lines of force. They run north-south and are pointing into the ground at an angle of the magnetic inclination. The extremes of field intensity are when the maximum number of lines are running through the flat sensor. Whether it is a minimum or a maximum depends on the direction in which the sensor is facing. The difference between the maximum and minimum is two times the magnetic field intensity of the earth. In the U.S., the magnetic field intensity is generally around 55,000 gammas. It ranges from 70,000 gammas near the magnetic poles to 30,000 gammas at regions near the equator. The field intensity is also affected by the ferromagnetic properties of the local rock, as well as electric currents in the ionosphere.

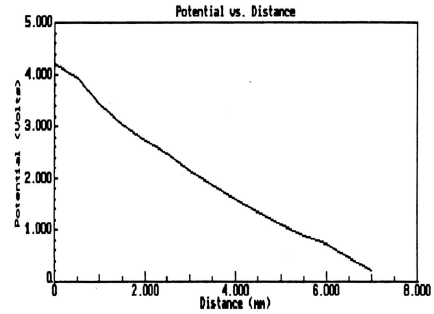
For more information on how geophysicists use magnetic field measurements to learn about the earth, refer to a text such as *Basic Exploration Geophysics*, C. Coruh and E. Robinson, 1988.

◊ Use the magnetic field sensor as a current sensor. The magnetic field produced by an electromagnet is proportional to the current flowing through it. Find or make an electromagnet with an iron core. Use tape or glue to attach the magnetic field sensor to one end of the magnet. It should be positioned perpendicular to the core of the magnet. The voltage from the Magnetic Field Sensor circuit will be directly proportional to the current flowing through the magnet. Adjust the gain and offset of the circuit so that you can use it over the range of currents you want to measure. Use the program MAGNET to record the voltage at two known current levels. With these two points you can figure out an equation that gives you the current through the electromagnet. You can change the program to display amps instead of voltage from the magnetic field sensor. Just change the line that displays the voltage to display the voltage multiplied by the conversion factor (current change/voltage change).

◊ Combine the sensor with a magnet to make a position sensor. This is one of the most common applications of magnetic field sensors. It provides an electrically-isolated, frictionless way to measure position. The sensor has no moving parts, and can be completely sealed to make it very durable. There are many possible ways to do this, depending on exactly what you need to measure. Here are some ideas:

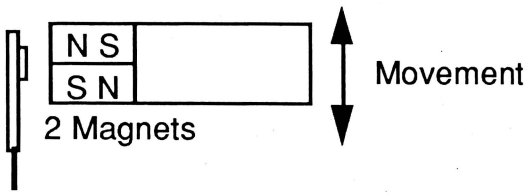
- Measure linear position using two magnets.

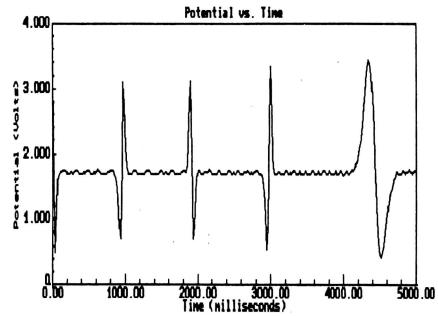




If the magnets are positioned correctly, the voltage is directly proportional to position, as shown in the graph above. This makes it easy to modify the MAGNET program to read position directly. The optimum position of the magnets depends on their intensities and the sensitivity desired. Using stronger magnets and positioning them close together gives a very high sensitivity over a very small range. Positioning the magnets farther apart gives a wider range with less accuracy. Experiment with different positions. Look for a spacing that gives a linear relationship as the two magnets are moved. You may want to change the program to make it graph your readings. This would make it easier to see the range over which it is linear.

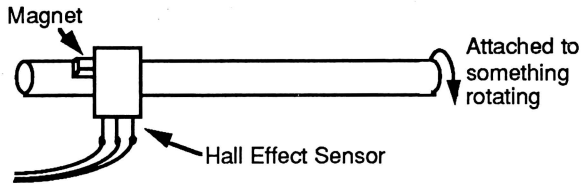
- Use two magnets to detect exactly when an object passes by the sensor. The following diagram shows how to do this:



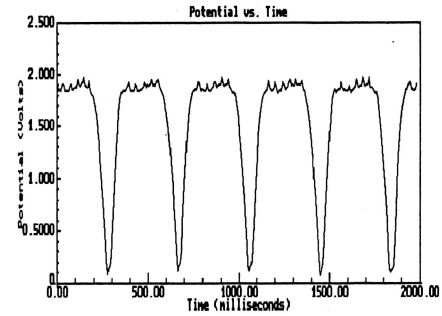


The magnets should be positioned so that the magnetic poles are in opposite directions. This gives a very sharp voltage change as the two magnets pass by the sensor. It will initially change as the first magnet approaches, then very quickly reverse that change as the other magnet becomes the dominant influence. The switch between the two magnets is very quick, as shown in the graph above. The rightmost pulse on the graph is the result of the magnet passing by more slowly, which shows the transition more clearly.

- Mount a magnet on something that rotates to measure RPMs—one possible application would be on an anemometer on a weather station. Not only is there no friction, but there are no moving parts (other than the anemometer itself) that could break down after long, continuous use and exposure to the elements. The sensor could be completely encased in sealant or plastic to protect it. The setup should be something like this:



Rotation Sensor



The sensor should be positioned close enough so that the computer reads a distinct voltage change each time the magnet passes by. The graph is of the voltage from the sensor using the setup shown above.

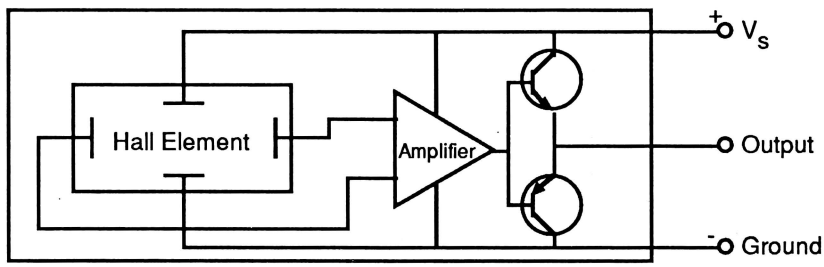
◊ Use one of the position sensors described above to measure force. You can mount a magnet on a bar of some kind and measure the force on that bar by determining the position of the magnet. Some commercial force transducers work on this principle. It is possible to measure a wide variety of forces by using different bars on which to mount the magnet. A small plastic bar would be very sensitive since it would bend very easily. A large steel I-beam could be used to measure very large forces.

◊ There are many other possibilities. Think about all the things that change position or have some moving part that could be used to activate the sensor. The sensor has no moving parts, and can be completely sealed to make it very durable.

Supplemental Information

The Hall Effect Transducer

The internal circuity of the Micro Switch SS94A1 Linear Hall Effect Transducer is diagrammed below:



Hall Effect Transducer

The transducer consists of a Hall effect sensor and an amplifier that linearizes and amplifies the signal. The supply voltage can be 4.5 volts to 8 volts. The output voltage will be one half of the supply voltage when no magnetic field is present. A magnetic field will cause this voltage to increase or decrease depending on the direction of the field. The transducer has a sensitivity of 3.125 mV/gauss when used with a 5-volt supply. The output is ratiometric to the supply voltage, so fluctuation in the power supply voltage will affect the output. The sensor handles the range of -400 to +400 gauss. Stronger fields will not cause a proportional change in voltage. One nice thing about the Hall effect transducer is that it cannot be damaged by intense magnetic fields.

The linearity of the transducer is guaranteed to be within 1%. The error introduced by changes in the temperature of the Hall effect sensor are specified as no more than 0.025% per degree C. The response time of the sensor is rated at 3 μs.

Micro Switch, and several other manufacturers, make linear Hall effect transducers that cover different flux density ranges. Digital Hall effect sensors are also available. These devices act as on-off switches, which trigger when the magnetic field reaches specific levels. Micro Switch is a Honeywell Division with distributors in major cities. For information on distributors, contact Micro Switch, Freeport, IL 61032, (815) 235-6600.

Program Design Notes

The MAGNET program on the disk will do the calculations necessary to convert the voltage readings from the Magnetic Field Sensor to magnetic field strength in gauss. Here is a section-by-section explanation of the program.

Lines 10 to 90: Introductory menu
This section asks which voltage input device you are using. It sets the variable VU$ to "A" if you are using the A-to-D Converter and to "V" for a Voltage Monitor.

Lines 100 to 990: Initialization
This section loads the appropriate machine-language program, and clears the screen and keyboard buffer.

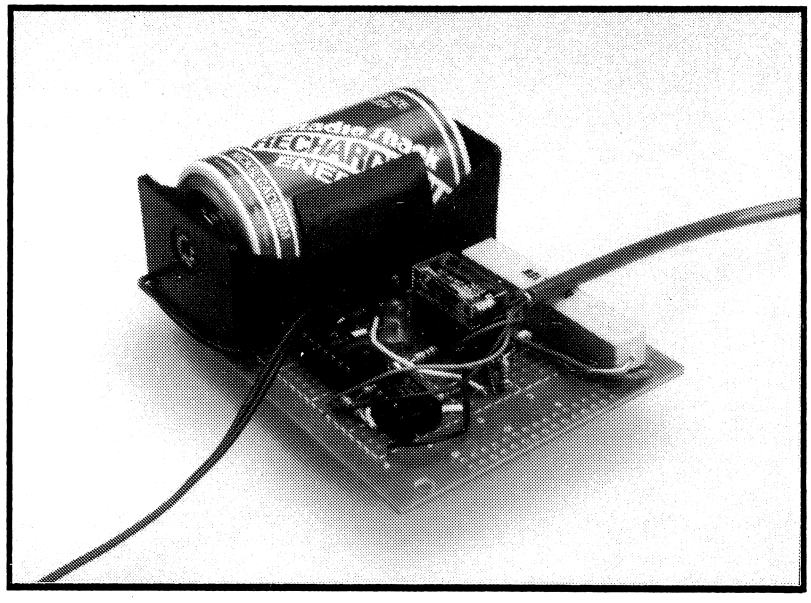
Lines 1000 to 1990: Display magnetic field readings
This section simply measures the magnetic field and displays the reading on the monitor.

Lines 10000 to 10990: Voltage check subroutine
These lines are used to do the check of the voltage from the voltage input device. It also converts from volts to magnetic field strength. This routine calculates the voltage slightly differently depending on the voltage input device being used.

The constant of proportionality used to convert volts to magnetic field is determined by the feedback resistance (the gain). The equation to use is B (Gauss) = (Voltage – Offset) / (Gain * 0.003125).

The following variables are used in the MAGNET program:

B	The magnetic field strength reading
C	Counts from the voltage input device
CH	The VIU channel to read
D$	Control D, used for disk access
F	Current frequency reading from voltage input device
GAIN	The gain of the magnetic field sensor—initially 10
GS	Temporary variable used for setting normal GS speed
KB	Value of character in keyboard buffer
OFFSET	The voltage offset of the magnetic field sensor—initially 2 volts
V	Present voltage reading
VU$	Voltage input device: "A" = serial A-to-D, "V" = voltage monitor or VIU



Project 9

Smart Battery Charger

Rechargeable nickel-cadmium batteries are widely used in many applications. For best results, they should be fully discharged before they are recharged. Smart Battery Charger does just that. It is "smart" in that it can sense the condition of a battery, discharge it fully, and then start the recharging process. It will also make graphs of the charging and discharging, so that you can check the condition of your batteries. You can make the circuit even "smarter" by adding a fast charge option.

Project Requirements

Voltage Input Device

You will need one of the following input devices:

- A-to-D Converter from Project 5 of this book
- Voltage Monitor from Project 7 of *How to Build a Better Mousetrap*
- Vernier Software 16-pin Voltage Input Unit
- Vernier Software 9-pin Voltage Input Unit (requires a one-line program change—see *Appendix F – Teacher's Guide*)

Computer

This project uses the annunciator output lines on the 16-pin Apple II game port. These lines are available on the II+, IIe and IIGS. This project cannot be done on other models of Apple II computers.

Background Knowledge

- Electronics: This is one of the most difficult projects in this book. The "Step-by-Step Instructions" are not as specific as on most of the projects. More of the details of component placement are left to you. Previous experience in assembling electronic circuits (for example, other projects in this book) is recommended. You should know how to read a schematic diagram. In addition, you will need to use Ohm's law to calculate the value of the charging resistor to use.
- Programming: No programming experience is required for the core project. Some of the extensions require experience with BASIC programming.

Parts

- ☐ 4N28 optoisolator, also known as an optical isolator or optocoupler (Almost any optoisolator with NPN phototransistor output or NPN photodarlington output will work. Do **not** use an optoisolator with "triac output" Here is a partial list of other part numbers we have used successfully: PS2002B, NTE3040, NTE3045, H11A1 and TIL111.
- ☐ 7404 Hex Inverter Integrated Circuit (Radio Shack #276-1802)
- ☐ 16-pin IC socket (Radio Shack #276-1998)
- ☐ 14-pin IC socket (Radio Shack #276-1999)
- ☐ 8-pin IC socket (Radio Shack #276-1995)
- ☐ 5-volt DPDT relay. (Some recommended suppliers and part numbers are: All Electronics #DRLY-52, Digi-Key #Z300-ND, or Newark Electronics #44F884)
- ☐ MPS2907, PNP transistors (Radio Shack #276-2023)
- ☐ (2) 1-amp rectifier diodes, PIV 50 volts or more (Radio Shack #276-1101 is a package of two)
- ☐ (1) yellow LED (Radio Shack #276-021)
- ☐ (1) green LED (Radio Shack #276-022)
- ☐ large circuit board (one suggestion is Radio Shack #276-168 or #276-162)
- ☐ (2) 330-Ω, 1/4 watt, 5% carbon resistor (orange, orange, brown, gold)
- ☐ 100-Ω, 1/4 watt, 5% carbon resistor (brown, black, brown, gold)
- ☐ 10-kΩ, 1/4 watt, 5% carbon resistor (brown, black, orange, gold)
- ☐ 4.7-kΩ, 1/4 watt, 5% carbon resistor (yellow, violet, red, gold)
- ☐ load resistor, 5 watt power resistor, either 5Ω or 1Ω (see text)
- ☐ charging resistor, value to be determined (see text), 1/4 watt, 5%
- ☐ battery holder (type depends on the type of batteries to be used)
- ☐ nickel-cadmium batteries to be charged
- ☐ miscellaneous jumper wires for connections on the circuit board

Tools and Supplies

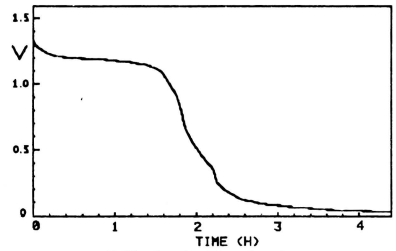
Soldering iron, solder, multimeter, wire cutters and wire strippers. You will also need a 5-volt power supply capable of delivering 100 mA or more. A meter that reads DC current up to a few hundred mA is helpful for testing the circuit. Finally, you will need some AAA, AA, C, or D nickel-cadmium batteries to recharge.

Background Information

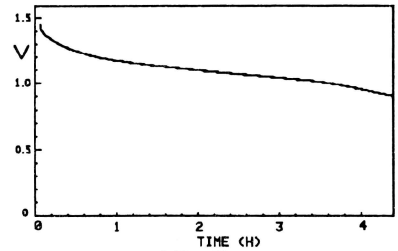
Nickel-cadmium batteries are used in electronic devices of all types. They are often called "Nicad" batteries, but that name is actually the trademark of one particular manufacturer, SAFT. Nickel-cadmium batteries can be recharged hundreds of times. Over time they can save their owners hundreds of dollars. An additional benefit is the reduction of waste in our landfills. The chemicals in disposable

batteries are often toxic. Some studies show disposable batteries to be one of the leading sources of heavy-metal pollution.

Each nickel-cadmium cell produces an electrical potential of about 1.3 volts when fully charged. Most other battery types produce about 1.5 volts per cell. When you buy a "AAA", "AA", "A", "C" or "D" nickel-cadmium battery, you may be expecting 1.5 volts of output, but you will get no more than 1.3 volts. This voltage difference is not normally a problem. The two graphs below show the discharge of a nickel-cadmium and an alkaline battery under the same load. There are several things worth noting about these graphs. The nickel-cadmium battery did not last as long as the alkaline battery. The nickel-cadmium battery holds very nearly its initial voltage for a long period and then the voltage drops quickly.



Nickel-Cadmium

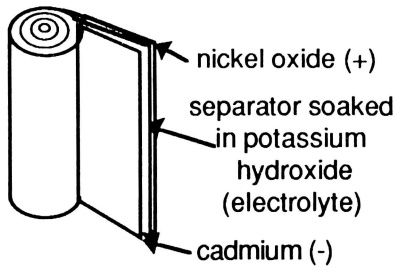


Alkaline

Battery Discharge Graphs

The alkaline battery starts at a higher voltage and the voltage drops steadily. When it still has a lot of life left in it, the alkaline battery's voltage has dropped to about the voltage of the nickel-cadmium battery. People designing electrical devices have to allow for battery voltages dropping quite a bit from the nominal 1.5 volts per cell, no matter what kind of battery is used. As a result, most electrical devices work well using nickel-cadmium batteries as a power supply.

Nickel-cadmium cells are made with a "jelly roll" construction as shown below:



The Inside of a Nickel-Cadmium Battery

The positive plates of the battery are made of nickel oxide (NiO). Cadmium (Cd) forms the negative plates. These two plates have a potassium-hydroxide saturated separator between them. All three layers are rolled up and placed inside a metal can. The can has a vent hole to release gases during charging and discharging.

The storage capacity of nickel-cadmium batteries is rated in milliampere-hours (mAh). This is simply an estimate of the maximum current this battery could supply steadily for a one-hour period. Typical ratings are shown below. The range of values for each size of nickel-cadmium battery results from small changes in the design of the battery.

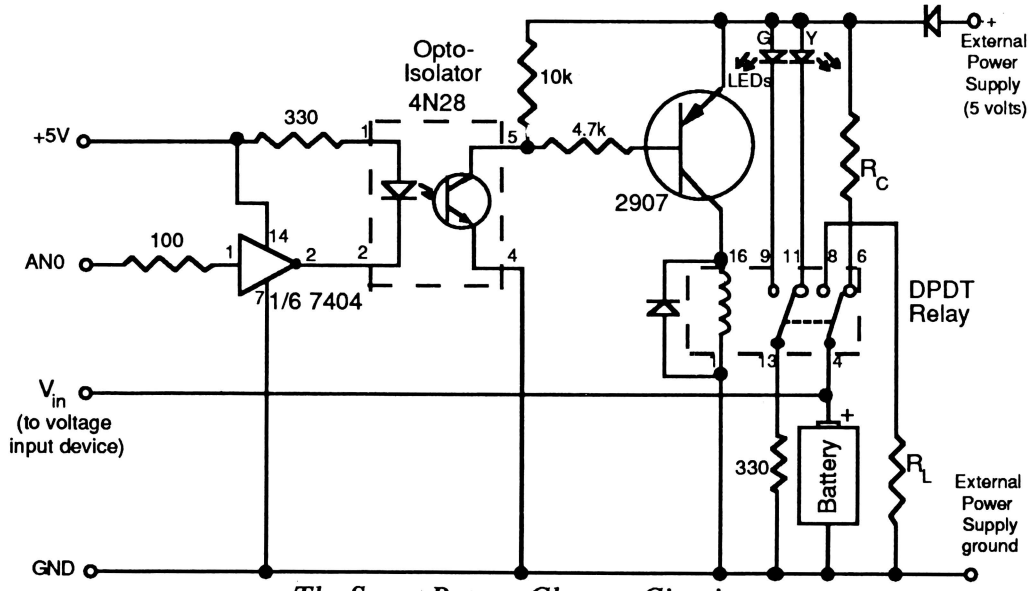
Battery Type	Storage Capacity (mAh)
AAA	50-200
AA	250-600
C	1000-2200
D	2000-4300

Many batteries have their capacity printed on the label. Remember that the mAh rating estimates the current that could be supplied for one hour. As a rough rule of thumb, the battery could supply half that much current for two hours, 1/10 as much current for 10 hours, etc. One of the unusual features of nickel-cadmium batteries is that they can also supply very large currents for short time periods. These batteries have very low internal resistance compared to most batteries. This makes nickel-cadmium batteries good for applications that draw a lot of current, such as drills, screwdrivers, etc. Never short the leads of a nickel-cadmium battery. The large current produced can cause extreme heating and can even cause the battery to explode.

Nickel-cadmium batteries are typically charged using a current numerically equal to one-tenth of the capacity (mAh) of the cell. For example, a AA battery with a storage capacity of 500 mAh would be charged with a current of 50 mA. A full recharge using this current requires 14 to 16 hours. Some nickel-cadmium batteries are designed for quick recharging in 4 to 6 hours or even faster. The label of any nickel-cadmium battery should clearly state the storage capacity and the suggested charging current and time.

For best results, nickel-cadmium batteries should always be discharged completely before they are recharged. If they are not completely discharged, they develop what is called a *memory effect*. If the batteries are constantly discharged only to a certain level, they will develop a memory for this kind of use and lose some of their capacity. Ideally, you should always discharge your batteries completely before recharging, but sometimes that is difficult to do. For example, if you are using a group of 4 or 5 cells together in an electronic device and it goes dead, it is unlikely that all of the cells are completely discharged. Some cells are stronger than others. Few people take the time to check and discharge each cell. Smart Battery Charger provides a way to automatically discharge the battery completely before recharging it.

Another consideration when using nickel-cadmium batteries in groups is that, to some extent, the set is only as strong as its weakest battery. One weak cell can drastically reduce the capacity of the entire set. For best performance, each set of batteries should be of equal storage capacity. If there is one weak cell, some of the stronger cell's energy is used in recharging the weaker cell. Smart Battery Charger provides a way to test batteries under a load to determine their capacity. Then you can prepare matched sets of batteries that will last longer.



The Smart Battery Charger Circuit

The complete schematic diagram for the Smart Battery Charger circuit is shown above. Most of the circuit is used to control a relay. The AN0 digital output line of the 16-pin game port is used as the control line. It can be set to either 5 volts or 0 volts by a program. The AN0 line is connected to a

7404 Inverter. This integrated circuit is used because it can safely "sink" more current than the Apple annunciator lines.[1] The IC is called an inverter because it changes the input signal it receives from low to high or vice versa. For example, if the AN0 line is at its high-voltage level, the output of the 7404 inverter (pin #2) will be a low voltage.

A key part of the circuit is the **optoisolator**. Optoisolators are small devices that look like integrated circuits. They include an LED (Light-Emitting Diode) and a phototransistor. In the circuit above, whenever the output of the 7404 is in its low state, current flows from the Apple's +5V line through the 330-Ω resistor and through the LED inside the optoisolator. The light produced by the LED affects the phototransistor inside the optoisolator. The phototransistor acts like a switch that is "closed" (conducting) whenever light strikes it, and "open" (not conducting) when no light is striking it. Since the phototransistor is sealed inside the case of the optoisolator, it can receive light only from the LED. There is no electrical connection between the two sides of the optoisolator. The only "connection" is by light. When the LED is on, the phototransistor conducts electricity.

When the phototransistor inside the optoisolator switches on, current flows through the 10-kΩ resistor, lowering the voltage applied to the base of the **transistor**. This transistor is now "turned on." Current can then flow through it and the **relay coil**. When current flows through the relay coil, it produces a magnetic field that switches the position of the **DPDT relay**. "DPDT"stands for *double pole, double throw*, which means that there are really two switches that move together and there are two positions for each switch. The **diode** near the relay is used to protect the transistor from electrical damage caused by the momentary backward current that occurs each time the relay coil shuts off.

This may seem like a lot of trouble to control a pair of switches. It is, but this circuit gets the job done and virtually eliminates any possibility of damaging your computer.

One of the switches in the relay is used to turn on one of two **LEDs**. The green one (for discharging) is switched on when the relay coil is activated, and the yellow one (for charging) is switched on the rest of the time. The **330-Ω resistor** limits the current through the LEDs. The other switch on the relay does the real work. When the relay coil is activated, it connects the battery to a low resistance **load resistor** for discharging the battery. The rest of the time it connects the battery to the external power supply via the charging resistor. This provides the charging current to the battery. This current is small enough that leaving the battery in place and charging it indefinitely with this current will not harm it. The **diode** in the charging circuit simply protects against reversed voltage being applied to the battery.

The Smart Battery Charger is designed to work with a voltage input device such as the Analog-to-Digital Converter, Project 5 in this book. A Voltage Monitor from *How to Build a Better Mousetrap* can also be used. The voltage input device allows the Smart Battery Charger system to monitor the status of the battery. Smart Battery Charger is "smart" because it can make decisions about what to do with a battery after checking its voltage.

The Smart Battery Charger circuit needs to be customized for the type of batteries you will be charging. The battery holder, load resistor, and charging resistor you use will depend on the type of battery. Suggestions for making these decisions are given in the "Step-by-Step Instructions."

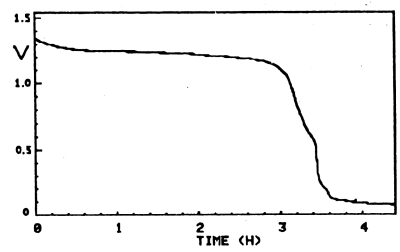
The BATTERY program on the *Chaos in the Laboratory Programs #2* disk is a complete battery maintenance program. It offers the following options:

V- CHECK VOLTAGE: This option simply reports the battery's voltage.

---

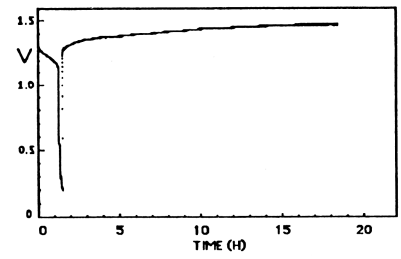
[1]The 7404 Inverter is a standard TTL device. Its outputs can safely sink a current of 16 mA. We use that current to turn on the LED inside the optoisolator. Connecting the optoisolator directly to the AN0 output line of the computer would not be a good idea, since it could put too much load on the line and damage the computer.

L - LOAD TEST: This option can do a load test on a charged battery and graph its voltage vs. time so that you can determine the battery's capacity. A sample load graph with a charged AA battery is shown here. Graphs made using this option cover about four hours. The current flowing at any time can be calculated from Ohm's law if you know the load resistor you are using: $i = V / R_L$. The area under the graph divided by the load resistance is the capacity of the battery. You might want to calculate these values and compare them to the published ratings of the batteries. This also provides an interesting way to compare various types of batteries.



A Sample Load Test

A - AUTOMATIC (DISCHARGE, THEN CHARGE) This option will check the status of a battery, discharge it completely if it is already charged, and then recharge it. A graph of the whole procedure is made at the same time. This sample graph shows a fully-charged nicad discharged and then recharged overnight. The graphs made using this option cover about 22 hours. Use this option for your routine battery charging.



An Example of the Automatic Mode - Discharge and Then Charge

G - DISPLAY PREVIOUS GRAPH: This option displays the last graph that was plotted.

P - PRINT GRAPH: This option allows you to print the previous graph if you have any one of the following printer configurations: (1) An ImageWriter printer; (2) A Grappler interface card and any dot-matrix printer; or (3) A PKASO/U interface card and any dot-matrix printer.

C - SET FOR SLOW CHARGE: This option sets the relay so that the battery charges. These last three options are used mainly for testing the relay circuit.

D - SET FOR DISCHARGE: This option sets the relay so that the battery discharges.

F - SET FOR FAST CHARGE: This option is useful only if you have added the fast charge circuit to the project as a project extension.

One feature of the Smart Battery Charger circuit is if you disconnect it from the computer or turn your computer off, the charger can still be used. As long as you keep the external power supply connected, the circuit will remain in the slow charge mode. This means you can check out a battery with the circuit, start it charging, and then use your computer for other jobs or turn it off while your battery charges. Later you can start up the battery program and check on the status of the charging.

The project extensions for this project include adding a fast-charge circuit and modifying the circuit to allow you to recharge nickel-cadmium batteries used in 9-volt applications. You can also do studies comparing various brands and types of batteries under different conditions.

Warning: Do not try to use the Smart Battery Charger with anything other than nickel-cadmium batteries. The battery should say "rechargeable" on the label. Batteries that were not designed for recharging can overheat or even explode if you attempt to recharge them.

Step-by-Step Instructions

We recommend a Radio Shack General Purpose IC PC Board (Radio Shack #276-150) for this project. You can use a number of different circuit boards—just make sure that it is big enough for all the components. If you plan to do some of the project extensions, you may want to use an extra large circuit board for adding features to the circuit.

☐ 1. **Experiment with component placement**. Before you solder anything, take a few minutes to experiment with the placement of the major components onto the circuit board. An efficient layout can save a lot of wiring and time. Also, you do not want to get halfway through the soldering and find there is not enough room on the circuit board you have chosen. Basically, you need to have room for one 14-pin socket (for the 7404), one 16-pin socket (for the relay), and one 8-pin socket (for the optoisolator). In addition you need a small area for the transistor. The few resistors and diodes usually can be fit in between these major components. The load resistor will be fairly large, so be sure to leave room for it. You may also want to mount your battery holder on the circuit board. Right now, you might want to skim through the instructions and diagrams that follow and position the key components. Look at the diagram of the complete circuit in step #9.

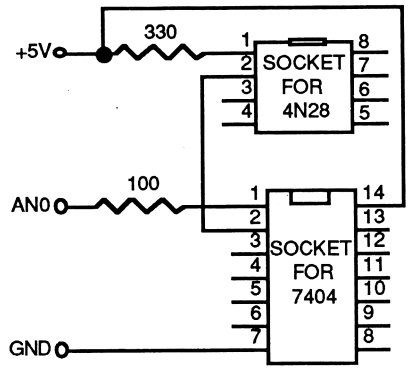
☐ 2. **Solder the 14-pin socket to the circuit board**. Place the 14-pin socket on the circuit board and solder all of its pins. Note which end of the socket is marked with a notch for pin #1 of the 7404 IC.

☐ 3. **Add an 8-pin socket nearby for the optoisolator.** Place an 8-pin socket on the circuit board for use with the optoisolator. The optoisolator will later be plugged into the socket, leaving holes 4 and 5 empty.[2] Solder the socket in place.

☐ 4. **Make connections to the voltage input device.** This project must be used together with a voltage input device (either Project 5 of this book, a Voltage Monitor from *How to Build a Better Mousetrap*, or a Voltage Input Unit). It uses three game port lines (AN0, GND, and +5V) and it must connect to the V_{in} terminal of the 5-pin DIN plug. We have found the best way to make these connections is to open up your voltage input device and make the four connections (V_{in}, AN0, GND, and +5V) inside. Run a four-conductor cable out of your voltage input device which will connect to your Smart Battery Charger circuit board. Label these four leads and put your voltage input device back together.

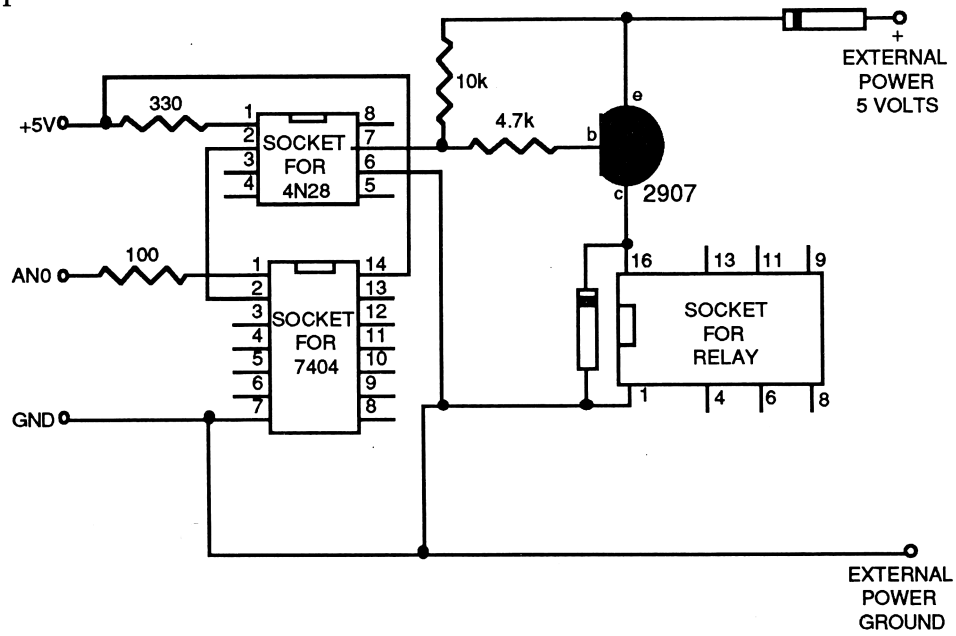
☐ 5. **Assemble the "computer side" of the circuit as shown below.** Three of the leads from the voltage input device connect to the circuit board. The 4th lead (V_{in}) will be connected to the battery "+" terminal in a later step. Add the two resistors and jumper wires to build the circuit shown. The 100-Ω resistor is used on the AN0 line as a safety feature to prevent damage to the Apple II if you accidentally short the AN0 line to ground as you are using the project. The 330-Ω current-limiting resistor connects the +5V lead to pin #1 of the optoisolator. You will need jumper wires to complete these connections.

[2]You may be wondering why we do not recommend using a 6-pin socket here, since the optoisolator has only 6 pins. 8-pin sockets are the smallest normally sold.



Working Diagram of "Computer Side" of the Circuit Board (Top View)

☐ 6. **Connect the relay coil circuit.** Six new components are needed to complete the "relay side" of the circuit: a transistor, a 16-pin socket, 2 diodes and 2 resistors. The working diagram below shows the connections. This circuit provides power to the coil of the relay, which will later be connected to pins #1 and #16 of the socket.



Working Diagram with Relay Coil Circuit Added

When placing the components on the board for this part of the circuit, be careful about the following:

◊ Be sure to orient the transistor properly. The three leads (**c**ollector, **e**mitter and **b**ase) of the transistor are labeled on the diagram above. If you are using a 2907 transistor in a plastic case with one flat side, the working diagram can be used to determine the proper orientation for that transistor. If your 2907 transistor is in another type of package, consult the data sheets to identify the leads.

◊ Be careful with the orientation of the diodes. The end with the band on it is the cathode "–" end. It connects to the transistor in both cases.

◊ The pin numbers shown for the 4N28 optoisolator socket correspond to the pin numbers of the 8-pin socket, not the 6-pin IC, which will later be placed in the socket.

Place all of these components on the circuit board. Use short jumper wires to make connections wherever needed. Double-check for the proper orientation of components and wiring connections, and then solder this part of the circuit together. When soldering the transistor, use a heat sink on the leads to prevent overheating.

☐ 7. **Determine the value of R_L for the batteries you plan to use.** The Smart Battery Charger circuit needs to be customized for the type of batteries you will be using. A large power resistor must be used here. The load resistor determines how fast your batteries will be discharged by the system. The discharge current will be the battery voltage (usually about 1.2 volts) divided by the load resistance. We recommend the following values:

For AAA and AA batteries, use 5 ohm, 2 watt or larger resistors.
For C and D batteries, use 1 ohm, 5 watt or larger resistors.

Try these values and then adjust the value if you want to after experimentation. Larger resistances will reduce the discharge current, and slow down the discharging process. Do not reduce the load resistance below 1Ω, or you may have some serious overheating problems. Do not use a small, low wattage resistor for the load resistor—it will be destroyed by overheating.

☐ 8. **Determine the value of R_c for the batteries you plan to be charging.** Check the battery's label. It usually lists the suggested charging current. If it lists both a "fast charge" and a "slow charge" current, use the "slow charge" value.

To calculate the proper resistance for the charging resistor (R_c), use the equation below. It is really just Ohm's law. The voltage drop across the resistor is the external power supply voltage (assumed to be 5 volts) minus the voltage drop across the diode (about 0.7 volts) and the voltage of the battery during charging (approximately 1.3 volts).

$$R_c = V / i$$

$$R_c = (5 - 0.7 - 1.3) / i$$

$$R_c = 3\ V / i$$

where i is the suggested charging current specified for the battery in amperes. R_c is in ohms. For example, many AA batteries suggest a 45 mA current for a slow charge (14-16 hours).

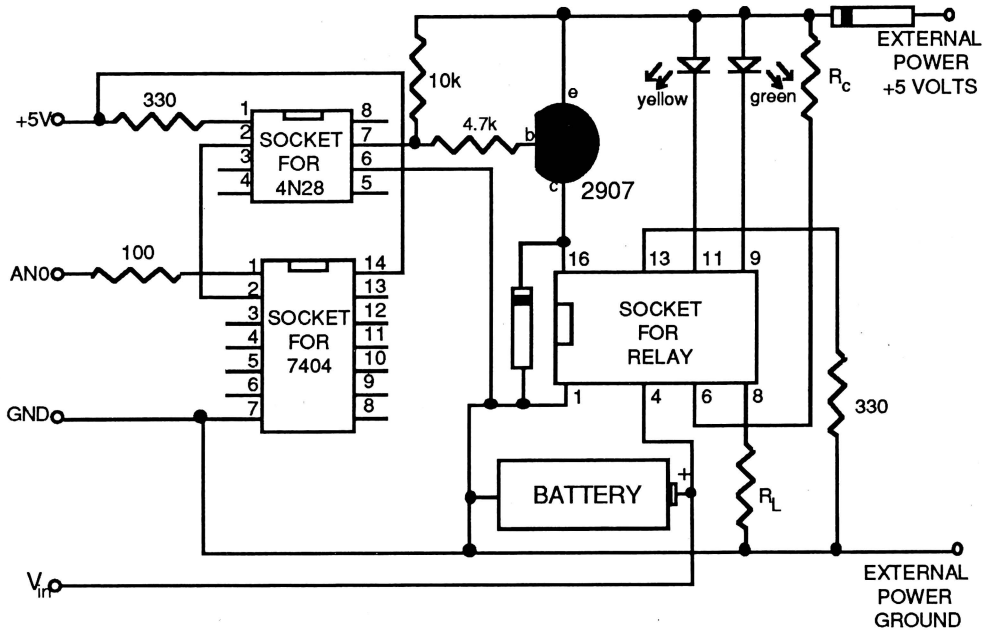
$$R_c = 3\ V / i = 3\ V / 0.045\ A = 67\ \Omega$$

In this case use a resistor of about 67 Ω. Use the nearest value you can find—68 Ω in this example.

Note: It is important not to use too small a resistor here. The Smart Battery Charger software and hardware are designed with the assumption that the resistor used here limits the current to a safe "slow charge" level for the battery being used. This current can be applied to the battery for an unlimited time without damaging it.

☐ 9. **Connect the charge and discharge circuits.** The working diagram below shows the complete circuit board with the charging and discharging circuits added. Remember that the load resistor should be a large power resistor to avoid overheating. Also, the LEDs must be positioned with their cathode side (the flat side) toward the relay.

At this point, you also need to solder the V_{in} lead from the voltage input device to the positive battery terminal.



Complete Working Diagram of Smart Battery Charger

☐ 10. **Plug the components into the IC sockets**. Plug the 7404 Open-Collector Inverter into the 14-pin socket. Be careful to orient pin #1 properly. Plug the optoisolator into the 8-pin IC socket. Place pin #1 of the optoisolator into hole #1 on the socket. Notice that the bottom two holes in the 8-pin socket are not used. Plug the relay into the 16-pin socket.

☐ 11. **Test the relays and the LED circuits.** Connect the external 5-volt power supply to the circuit and turn it on. With the computer turned off, plug the ribbon cable into the 16-pin game socket. Place the *Chaos in the Laboratory Programs #2* disk in the disk drive and turn on the computer. If the computer does not start up normally, turn it off and check for a mistake in the wiring. Otherwise, make a quick check of the circuit by looking, smelling, and feeling for anything hot. The yellow LED should be on. None of the components should be warm. If they are, there is a problem with the circuit.

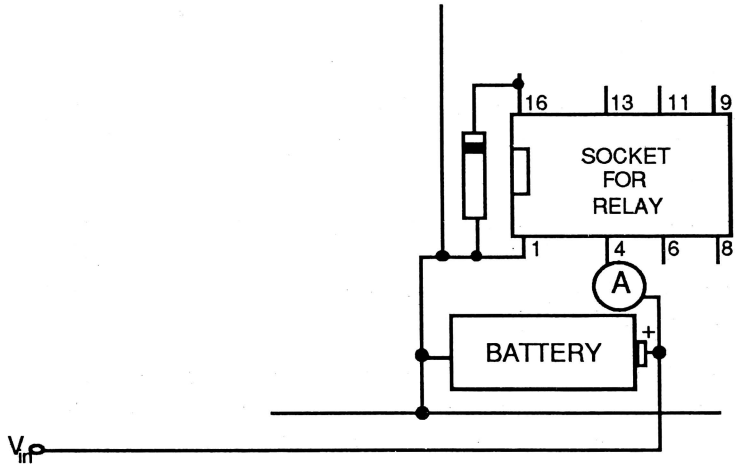
If everything is cool (literally), run the program BATTERY. If you are using the A-to-D Converter (Project 5 of this book), select "A" at the initial menu. If you are using a Voltage Monitor, select "V". Then try pressing "D" and then <RETURN>. This should turn on the green LED. Pressing "C" and <RETURN> should turn the yellow LED back on. Only one LED should be on at a time.

If the LEDs do not operate as expected, shut off the computer, disconnect the external power supply, and double-check the circuit. If you still cannot find the problem, refer to the "Troubleshooting" section below for suggestions on locating the problem.

☐ 12. **Put a battery in place and test the charge/discharge circuits.** After you have the LEDs working correctly, it is time to put the battery in place. With the battery in the holder, the external power supply on, and the program running, choose C - SET FOR CHARGING from the menu. Now choose V - READ VOLTAGE. The program should start displaying the voltage of the battery as it is being charged. This is always 1.2 volts or more, even if the battery is not fully charged. If the voltage is not at least 1.2 volts, the charging circuit is not working correctly. Double-check it.

Now press <RETURN> to stop the voltage measurement and choose D - SET FOR DISCHARGE from the menu. Choose V - READ VOLTAGE again. The program will now display the voltage across the battery as it is discharged through the load resistor. The voltage will depend on the charge of the battery. It will start to drop at least slightly in a few minutes.

☐ 13.**(optional) Check the charge/discharge circuits using a milliammeter.** If you have a meter that will read current in milliamps, you may want to build it into the circuit temporarily as shown below. Make sure that the meter can handle currents as large as the discharge currents you plan to use. You need to place the ammeter in series with the battery. It becomes part of the circuit between pin #4 of the relay and the "+" terminal of the battery.



How to Add a Milliammeter to the Circuit for Testing

The ammeter is helpful to make sure that the proper charging and discharging currents are being used. Note that charging currents will be positive and discharging currents will be negative. Compare the charging current with the recommended slow charge current. The discharge current should be approximately equal to the 1.2 volts divided by the resistance of the load resistor. The current will drop as the battery loses it charge.

☐ 14. **Use the Smart Battery Charger Load Tester.** If you have a battery that is already charged, and you want to test its capacity, put it in the battery holder and select L - LOAD TEST from the menu. The load resistor will be connected across the battery, and a graph of voltage vs. time is made. If you have several batteries of the same type, you can determine which batteries are in the best condition. Since you know the load resistor resistance, you can even calculate the battery's capacity in amp-hours. The area under the voltage vs. time graph (in units of volt-hours) divided by the load resistance is equal to the capacity in amp-hours.[3]

☐ 15. **Use the Smart Battery Charger Automatic Mode.** If you want to charge a battery, select A - AUTOMATIC (DISCHARGE, THEN CHARGE) from the menu. The Smart Battery Charger will connect the load resistance and begin to monitor and graph the battery's voltage using your voltage input device. The program will stay in the discharge mode until the battery is discharged to 0.2 volts. At this point, the program will switch the relay and the charge circuit will be connected. The charging will continue indefinitely. This is not harmful to the battery. During both the discharging and the charging, a graph of voltage vs. time is made.[4]

Remember that Smart Battery Charger should be used only with nickel-cadmium batteries that are designed to be recharged.

[3]The discharge current is equal to the voltage divided by the load resistance; therefore, the area divided by the resistance is equal to the capacity in amp-hours.

[4]During charging, the program may display FAST CHARGE or SLOW CHARGE. This is a feature of the program to support the fast recharge project extension. There is no difference between FAST CHARGE and SLOW CHARGE unless you build the fast charge project extension.

Troubleshooting

The following steps should help you locate a problem in the circuit. Proceed through these checks in order until unexpected results are encountered. This should help you zero-in on the source of the problem. These tests assume that you have the circuit set up for normal operation. Attach a voltmeter's negative lead to the Apple computer's GND lead (pin #7 of the 7404). Attach the circuit to the game port and turn on the computer. Run the BATTERY program. Turn on the external power supply.

1. Is the circuit getting power? Touch the positive lead of the voltmeter to pin #14 of the 7404. The voltage here should be approximately 5 volts. If not, you probably have a mistake in the cable to the voltage input device.

2. Is the LED in the optoisolator turning on? Connect the positive lead of the voltmeter to the middle pin on the left side of the optoisolator. The voltage at that point should change from near 0 when charging to about 4 volts when you change to discharge. If the voltage does not change, there must be something wrong with the 7404 circuit.

3. Is the external power being delivered to the circuit? Touch the positive lead of the meter to the top pin of the transistor (emitter). The voltage at that point should be about 4.2 volts. If not, there is a problem with the external power supply, its connections, or the diode.

4. Is the phototransistor inside the optoisolator turning "on"? Touch the positive lead of the meter to the middle pin of the right side of the optoisolator. The voltage at that point should change from near 4.2 volts charging to near zero volts when discharging. If it does not, the phototransistor is not wired properly.

5. Is the 2907 transistor turning on? Touch the positive lead of the meter to the bottom lead (collector) of the transistor. The voltage at this point should change from near zero when charging to about 4 volts when you change to discharge. If there is no change, double-check the wiring of the transistor. If this test fails, and the wiring is correct, try substituting a new transistor.

Project Extensions

Circuit Modifications

CONVERT THE CIRCUIT FOR USE WITH "9V" BATTERIES

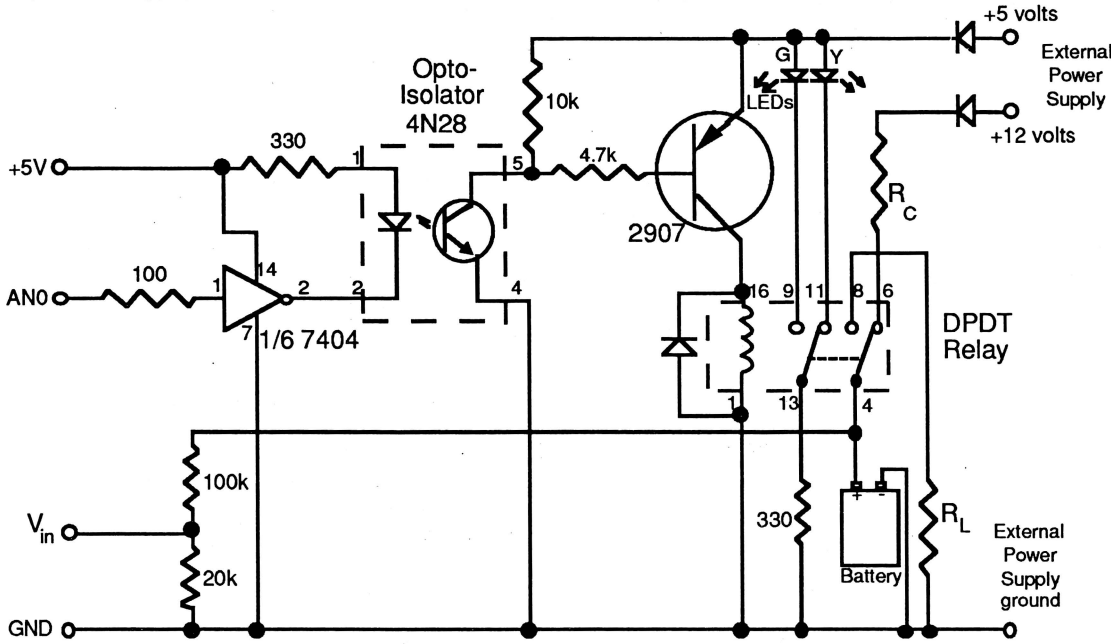
The Smart Battery Charger circuit is designed for use with 1.3-volt batteries. By making a simple change in the circuit, you can use it to maintain batteries that are used in "9-volt" applications. Most nickel-cadmium batteries sold for use in "9-volt" applications are really six 1.3-volt nicad cells in series. The circuit below has four major changes from the core project: (1) The voltage used for charging this time must be higher (12 volts). (2) In order to allow the BATTERY program to be used without modification, a voltage divider circuit is included so that the voltage input device reads 1/6th of the total voltage of the battery. (3) These batteries should be discharged through loads with higher resistance. A typical load resistor should be about 50 to 100 Ω. (4) These nickel-cadmium batteries use much smaller currents for charging. To calculate the proper resistance for the charging resistor (R_C), use the equation below. This calculation assumes you will be using a 12-volt external power supply to charge the battery.

$$R_C = V / i$$
$$R_C = (12 - 0.7 - 7.8) / i$$
$$R_C = 3.5\ V / i$$

where i is the suggested charging current specified on the battery in amperes. R_C is in ohms. For example, many nicad batteries designed for use in 9-volt applications suggest a 7 mA current for a slow charge (14-16 hours).

$$R_C = 3.5\ V / i = 3.5\ V / 0.007\ A = 500\ \Omega$$

A schematic for a "9-volt" battery charger is shown below. Notice that this circuit uses a 5-volt external power supply for the relay and LEDs and a 12-volt power supply for the charging current.



A "9-Volt" Smart Battery Charger

ADD A "FAST CHARGE" CIRCUIT

Many nickel-cadmium batteries specify "fast charge" conditions that can be used for recharging. For example, certain "AA" cells specify a fast charge current of 150 mA for 4 hours. This compares with the "standard charge" conditions of 45 mA for 14 hours. The larger current is fairly easy to supply just by reducing the resistance in the charging circuit. This larger current must be used with caution, however, because if the battery is charged for too long with the larger current, it can be destroyed. The battery charger must be able to sense when it is time to reduce the charging current.

The circuit below adds a second relay controlled by the AN1 digital output line of the game port. This relay can connect the battery to the charging voltage though a second resistor (R_f). This resistor is in parallel with the original charging resistor (R_c) and therefore reduces the total resistance (R_t). The result is an increase in the charging current. The size of the total resistance you want for fast charging can be calculated from:

$$R_t = (V_c - 0.7\text{ V} - 1.3\text{ V}) / (\text{fast charge current})$$

You then need to determine what resistor can be added in parallel with your present charging resistor (R_c) to give you the total resistance you need. To do this, solve the equation below for R_f:

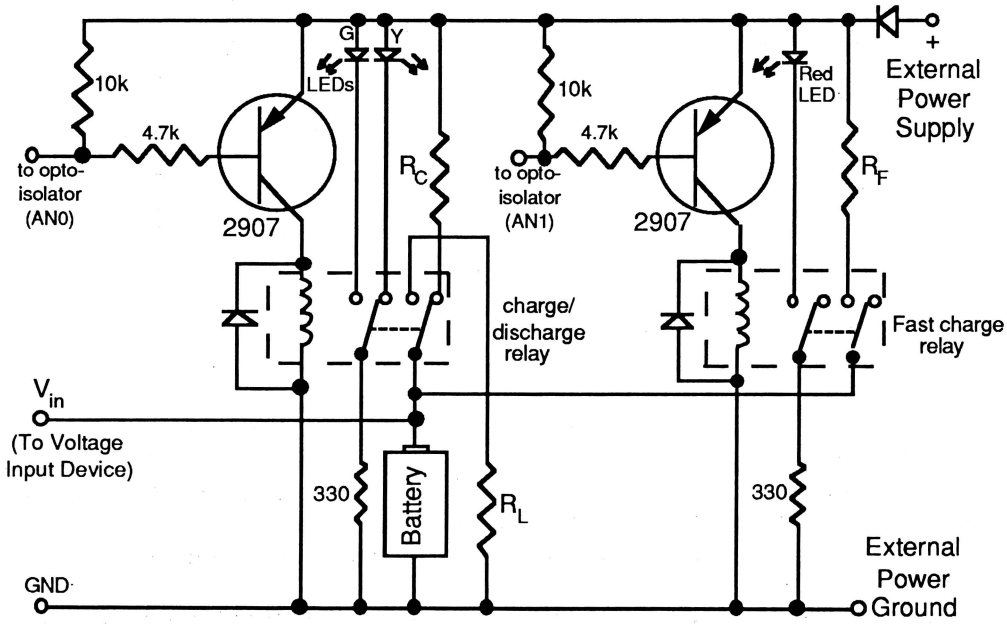
$$R_t = \frac{R_c * R_f}{R_c + R_f}$$

This is the equation for the total resistance of R_c and R_f in parallel.

Some typical values are shown below. Always use the charge currents suggested on the outside of the battery if possible. This table gives typical fast charge currents and the resistor values you would use to produce them. It assumes that you are using a 5-volt external power supply for charging the batteries.

Battery Type	Fast Charge Current (mA)	R_t (Ω)	R_c (Ω)	R_f (Ω)
AAA	45	67	188	104
AA	150	20	67	28
C	330	9	30	13
D	330	9	30	13

The complete schematic of the Smart Battery Charger with the fast charge circuit added is shown below. The BATTERY program on the *Chaos in the Laboratory Programs #2* disk supports the fast charge circuit. The F - SET FOR FAST CHARGE option on the menu activates it. Also, the A - AUTOMATIC (DISCHARGE, THEN CHARGE) option will activate the fast charge circuit as soon as the battery is discharged. Fast charge will be used until the battery voltage reaches 1.5 volts. Then the fast charge circuit will be turned off and the battery will be left in the slow charge mode.



MODIFY THE SMART BATTERY CHARGER FOR USE WITH SEVERAL TYPES OF BATTERIES

Each different type of nickel-cadmium battery requires a different charging resistor, load resistor, and battery holder. If you are clever, you can probably design a connector system so that different battery holders and different resistors can be quickly put in the circuit.

Program Improvements

The BATTERY program on the *Chaos in the Laboratory Programs #2* disk is fairly simple, but it can be improved by adding new features. The "Program Design Notes" section below explains how the program operates. It should help you in making modifications. Some suggested changes are given below:

◊ If your computer has a "clock card" to keep track of real time, change the program so it reads and uses the clock time. This will make the horizontal axis of the graphs more accurate.

◊ Add current to the display during discharge. Once you decide on the value of the load resistor, you can change the BATTERY program so that it displays the current used during discharge. The discharge current can be calculated as follows: $i = V / R_L$

◊ Add current to the display during charging. Once you have decided on the value of the charging resistor and the power supply voltage, you can change the BATTERY program so that it displays the current used during charging. The charging current can be calculated as follows:

$$i = V / R$$
$$i = (V_s - 0.6 - V_b) / R_c$$

where: V_s = power supply voltage
V_b = battery voltage (measured by the computer)
R_c = charging resistor
the 0.6 volts is the voltage drop across the diode in the charging circuit

Add the following line to the BATTERY program, with the appropriate values used for R (in ohms) and VS (in volts): `2175 IF S$="SLOW CHARGE" THEN PRINT TAB (20); INT ((VS - 0.6 - V) / R * 1000); REM PRINT CURRENT IN MILLIAMPS.` Additional changes will be needed if you are using the fast charge project extension.

◊ Add a calculation of the capacity of a battery as it is discharged in the load test. The area under the curve in the load test divided by the load resistance is the capacity in amp-hours. Add instructions to the BATTERY program so that it calculates the area and the capacity. First the program should initialize a new variable to zero. Each time a new voltage is plotted, the program should add to this variable the new area under the graph added. When the whole graph is completed, the area should be divided by the load resistance to calculate the capacity in amp-hours.

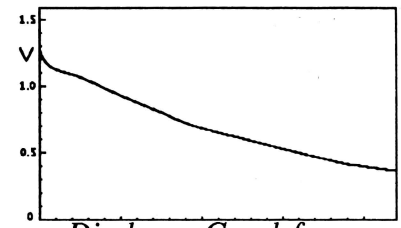
◊ Adjust the time delay values so that the timing of the program is more accurate. We have built delay loops into the BATTERY program so that it takes readings approximately once a second. This is used as a time base for the program. A small error in this timing will cause the times to be off over long time intervals. You may want to change the loop limits to make the timing more exact.

Additional Experiments

A number of different studies of batteries can be done with the Smart Battery Charger. Here are some topics for investigation:

- Are the expensive batteries worth the extra cost when you consider their capacities?
- Which batteries work best under heavy loads? Under light loads?
- Which batteries hold their charge best during storage?
- How much capacity does a battery lose after 100 discharge/charge cycles? After 1000 discharge cycles?

Even though Smart Battery Charger can be used to charge only nickel-cadmium batteries, it can be used to do load tests on other types of batteries. To do this, disconnect the charging resistor, place the battery to be tested in the holder, and run the load test mode of BATTERY. A graph of the discharge of the battery will be made. This allows you to expand your battery tests to include carbon-zinc and alkaline batteries. The graph of an alkaline battery discharge at the beginning of this chapter was made in this way. Here is a graph showing the discharge of an inexpensive carbon-zinc battery made in the same way.



Discharge Graph for a Zinc-Carbon Battery

Supplemental Information

Program Design Notes

The BATTERY program controls the Smart Battery Charger relay and reads the voltage from the battery. Here is a section-by-section explanation of the program:

Lines 10 to 90: Introductory menu
This section asks which voltage input device you are using. It sets the variable VU$ to "A" if you are using the A-to-D Converter, and "V" for a Voltage Monitor.

Lines 100 to 200: Initialization
This section loads the appropriate machine-language program, and clears the screen and keyboard buffer.

Lines 200 to 500: Menu

Lines 2000 to 2990: Automatic mode

Lines 3000 to 3990: Display voltage

Lines 4000 to 4990: Load test

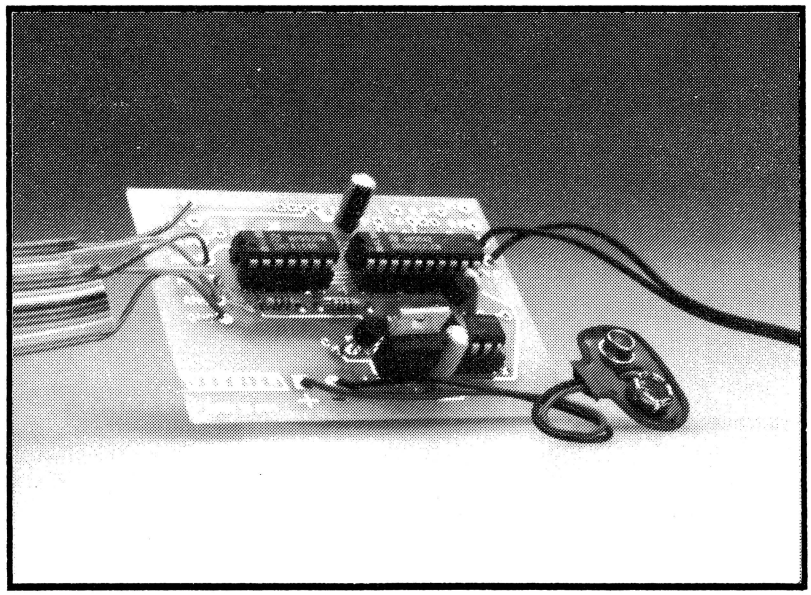
Lines 5000 to 5400: Print screen

Lines 10000 to 10090: Read voltage

Lines 20000 to 20900: Draws the background for the graph

The following variables are used in the BATTERY program:

C Counts from the voltage input device
CH The channel to read if a Voltage Monitor is used
D$ <Control> <D>, used for disk access
F Frequency reported by Voltage Monitor
GS Temporary variable used for setting normal GS speed
I, J, JJ, XX, YY Dummy loop counters
M$ Mode of operation
PF Printer format
PR Printer slot
PT Printer type
Q$ Dummy input string
S2 Printout size
S$ Charge/discharge status string
T Time (in seconds)
T1 Time to discharge to 1.2 V in automatic mode
T1$, T2$ Strings used to print horizontal label on graph
T2 Time in seconds to discharge to 0.2 V in automatic mode
T3 Time in seconds to charge to 1.4 V in automatic mode
T4 Time in seconds to charge to 1.5 V in automatic mode
TD Time in seconds per pixel on the graph
V Voltage
V1 Starting voltage in automatic mode
VU$ Voltage input device: "A" = A-to-D Converter, "V" = Voltage Monitor



Project 10

D-to-A Converter/Digitized Sound

In this project you will build a digital-to-analog converter that is controlled through the Apple game port. With this circuit the computer can control an output voltage between 0 and -2.5 volts in steps of 0.01 volts. The output signal can be used to control small motors, LEDs, or lamps. The circuit can be connected to a speaker system to allow you to play digitized sounds that are included on the *Chaos in the Laboratory Programs* disks. By combining this project with the A-to-D Converter (Project 5), you can digitize and play back sounds you record yourself.

Project Requirements

Computer

This project uses the annunciator output lines on the 16-pin Apple II game port. These lines are available on the Apple II+, IIe, and IIGS. This project cannot be done on other models of Apple II computers.

Background Knowledge

- Electronics: If this project is done using the Vernier Software D-to-A Converter Parts Kit, the project is fairly easy because a custom circuit board is included. If you are not using the custom circuit board, you will have to use the schematic diagram as a guide and use a lot of jumper wires. This makes the project fairly complicated.

- Programming: A BASIC program is supplied to operate the system. For some of the project extensions you may want to change the BASIC program. In some cases, you may even want to change the machine-language program.

Parts

- ☐ 74164 serial shift register IC
- ☐ DAC0832LCN serial digital-to-analog converter IC
- ☐ LF353 dual op-amp IC
- ☐ 2907 PNP transistor (Radio Shack #276-2023)
- ☐ LM7905 voltage regulator
- ☐ LM385 2.5-volt reference diode
- ☐ circuit board (Vernier Software D-to-A Converter custom circuit board)
- ☐ 20-pin IC socket (Radio Shack #276-1999)
- ☐ 14-pin IC socket (Radio Shack #276-1998)
- ☐ 8-pin IC socket (Radio Shack #276-1995)
- ☐ rectifier diode (Radio Shack #276-1101)
- ☐ 16-pin ribbon cable to the Apple game port (see *Appendix A*)
- ☐ (3) 100-Ω, 1/4 watt carbon resistor (brown, black, brown, gold)
- ☐ 22-kΩ, 1/4 watt carbon resistor (red, red, orange, gold)
- ☐ 10-kΩ, 1/4 watt carbon resistor (brown, black, orange, gold)
- ☐ 1-kΩ, 1/4 watt carbon resistor (brown, black, red, gold)
- ☐ (2) 10-μF electrolytic capacitor (Radio Shack #272-1013)
- ☐ 0.01-μF capacitor (Radio Shack #272-1065)
- ☐ 9-volt battery clip (Radio Shack #270-325)
- ☐ about 1 meter of two-conductor cable

Note: A D-to-A Converter Parts Kit containing all of the components needed for this project, including a custom-printed circuit board, is available from Vernier Software. This kit sells for $21.00 (Order Code CDA-16). An order form is provided at the back of this book.

Tools and Supplies

Low-power soldering iron, solder, wire cutters, wire strippers, a voltmeter, and an ohmmeter for testing.

Background Information

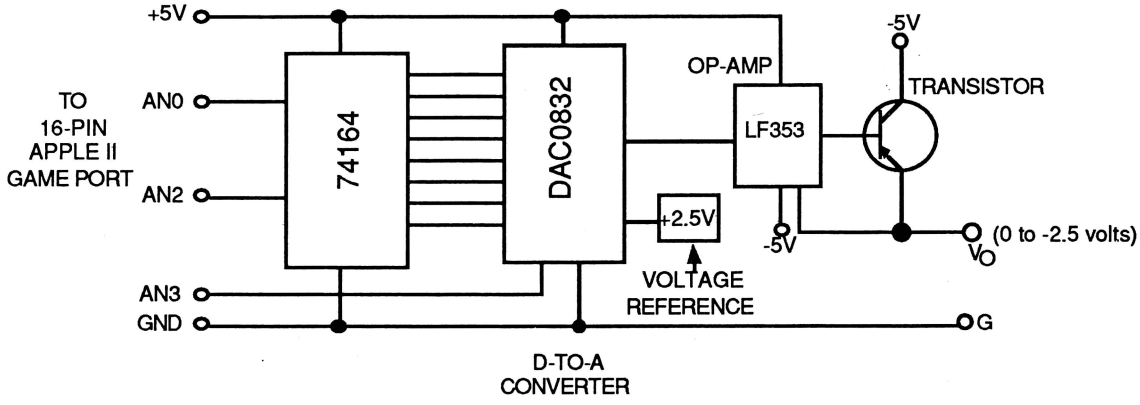
Computers like the Apple II are *digital* devices; that is, information used internally by the computer is always coded in the form of patterns of on or off signals. A digital data line can have only two possible values—on (usually near +5 volts) or off (zero volts). An example of digital data in the non-computer world would be a light with an on-off switch. There are only two possible settings. It is fairly easy to use a computer's digital output lines to turn on and off a lamp.

Often it is useful to get a digital computer to control an *analog* signal. An analog signal is one that can vary over a wide range of values. An example of an analog device would be a lamp with a dimmer control. There are many different possible settings for this lamp. How can a digital computer be used to do more than just turn a lamp on or off? The answer is by using a *digital-to-analog converter*. Digital-to-analog converters are integrated circuits made especially to take a pattern of digital information from a computer and produce a voltage out which varies over a number of possible values.[1]

---

[1]Note the comparison with analog-to-digital converters, which were discussed in Project 5. These two devices are essentially opposites. One translates digital information from the computer into an analog voltage, and the other translates a voltage into a digital number the computer can understand.

The standard way to connect a digital-to-analog converter to a computer is to plug it into one of the slots inside the computer. This allows the digital-to-analog converter access to all of the data lines of the computer for reading the digital data. In this project, our goal is to do digital-to-analog conversion using the Apple II game port. This a little more difficult, since there are not enough lines on the game port to send data directly to the digital-to-analog converter. We use an additional IC called a *serial shift register* to control the digital-to-analog converter. Two lines connected to the serial shift register allow us to control the necessary eight lines connected to the digital-to-analog converter. A simplified diagram showing how the circuit works is shown below:



A Block Diagram of the A-to-D Converter

The voltage out of the digital-to-analog converter IC will output any one of 256 values over the range 0 to -2.5 volts. This signal goes through a dual op-amp and a transistor, which buffer the signal and provide the ability to supply enough current to operate small electrical devices.[2]

One important thing to notice about the Digital-to-Analog Converter circuit is that its output is a negative voltage. When a digital value of 255 is sent to the circuit, the output is -2.5 volts. When the digital value is zero, the output voltage is 0 volts.

Once you have the digital-to-analog circuit working, what can you do with it? There are more project extensions for this project than for any other project in this book. For starters, the circuit can be used to control a motor, lamp, heater, and other electrical device. You can control the speed, brightness, or temperature of the device—not just turn it on or off. The program D.TO.A on the *Chaos in the Laboratory Programs #2* disk allows you to set the voltage level. Project extensions show how to change the voltage range and how to add a power transistor to control larger currents.

The Digital-to-Analog Converter can change its output very rapidly—up to 6000 times a second. As a result, you can use the circuit as a function generator. The program SAMPLE.PLAY allows you to output sine, square, and sawtooth wave forms. You can control the frequency of these waves.

By adding just a little circuitry to the circuit board and then connecting either headphones or amplified speakers to the Digital-to-Analog Converter, you can play sounds. The SAMPLE.PLAY program allows you to load previously digitized sounds and play them through the speakers. The backs of both of the *Chaos in the Laboratory Programs* disks are full of interesting sounds—themes from old TV shows, musical instruments, train sounds, etc. The program supports playing the sounds at different speeds or even backwards. If you are a programmer, you may want to build these sounds into your own programs.

The next step in digitized sound experiments is to add an A-to-D converter to the project so that you can digitize your own sounds. The circuit for Project 5 is used. With this circuit and a microphone, you can digitize several seconds of sound and save it on disk.

---

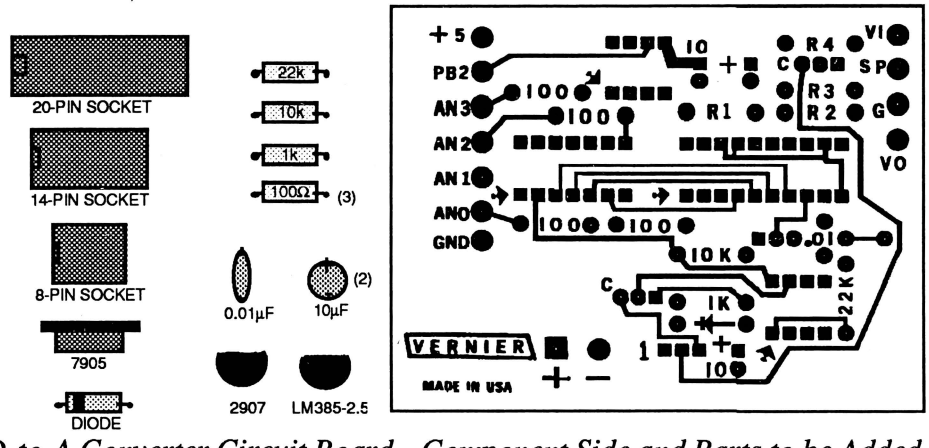
[2]The "Supplemental Information" section of this chapter has a more complete explanation of how this circuit works, including a complete schematic diagram.

Step-by-Step Instructions

Overview: These instructions assume that you are using a Vernier Software D-to-A Converter Parts Kit, which includes a custom circuit board. If you are building the project using a general-purpose circuit board or a prototyping board, you should use the schematic diagram in the "Supplemental Information" as a guide.

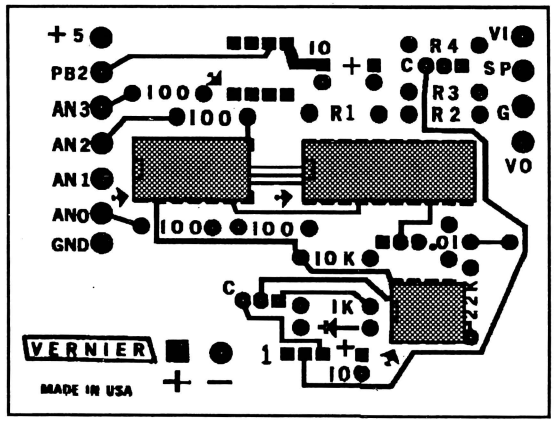
As you build this project, do not be concerned when you do not fill up the circuit board completely. Some of the holes in the circuit board are for project extensions that you may want to add later.

☐ 1. **Identify the parts.** The Vernier Software D-to-A Converter Parts Kit contains all of the components needed for this project. Check the parts list included with the kit and identify each part.

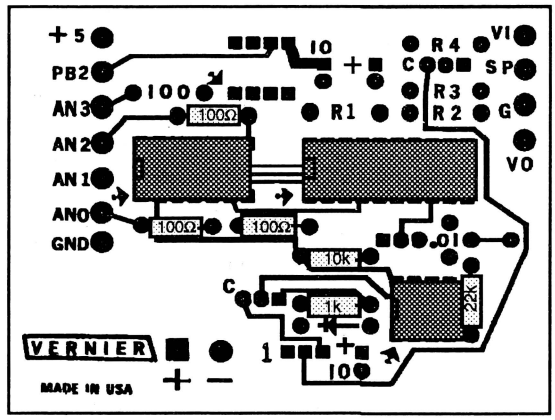


The D-to-A Converter Circuit Board—Component Side and Parts to be Added

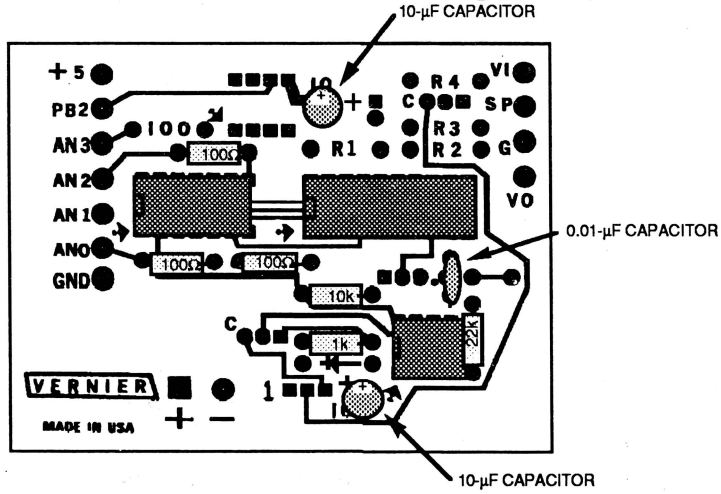
☐ 2. **Solder the sockets to the board.** Position the three sockets in the positions shown below. All components go on the side of the circuit board with the lettering on it. The 14-pin socket goes toward the left side, the 20-pin socket on the right side and the 8-pin socket near the lower right corner. Note the notches in the sockets. These are meant to mark where pin #1 of the IC will be plugged in later. The notches all should point toward the left. Solder all of the pins of all three sockets to the back side of the circuit board.



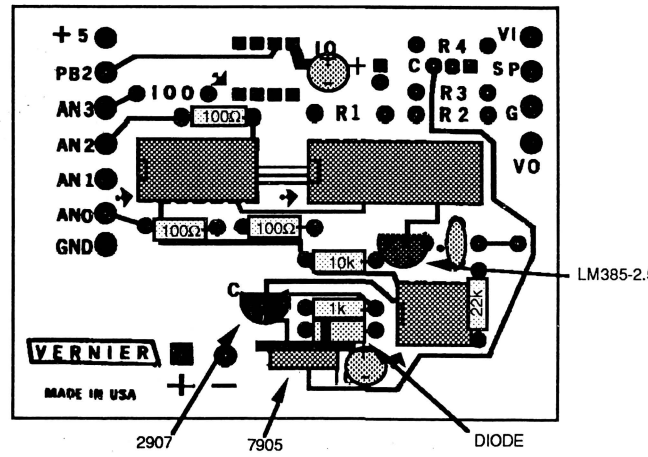
☐ 3. **Add the resistors to the circuit board.** There are six resistors used in this circuit. Place them as shown in the diagram below. The circuit board has the value for each resistor labeled on the component side. Push the leads into the holes as far as possible. Solder all of the leads. Cut off the excess leads as close to the circuit board as possible.



☐ 4. **Add the capacitors to the circuit.** There are three capacitors to be added. The two 10-µF capacitors are polarized—one side has a "–" mark on it. That side goes toward the bottom of the circuit board. The other capacitor (0.01 µ-F) is not polarized. Solder all three capacitors in place. Cut off the excess capacitor leads as close to the circuit board as possible.



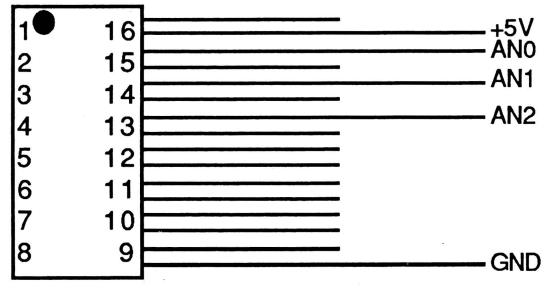
☐ 5. **Add the voltage regulators, transistor, and diode.** Each of these components must be oriented in a particular way. The LM385-2.5 and the 2907 transistor both should be positioned with all three of their leads in a row of horizontal holes and their flat side pointing toward the top of the circuit board. When soldering these two components, use a heat sink to prevent damage from overheating.



The diode has a stripe on one end. That end goes toward the left. The 7905 regulator should have three leads in a row of horizontal holes, with the metal side toward the top of the circuit board. Solder these two components in place. A heat sink is not needed on either component. Cut off the excess leads from all the components added in this step.

☐ 6. **Prepare the 16-conductor ribbon cable that will connect the circuit board to the computer.** Connections for the AN0, AN1, AN2, +5V and GND leads are shown below:

Pin 1 = +5V
Pin 8 = GND
Pin 15 = AN0
Pin 14 = AN1
Pin 13 = AN2

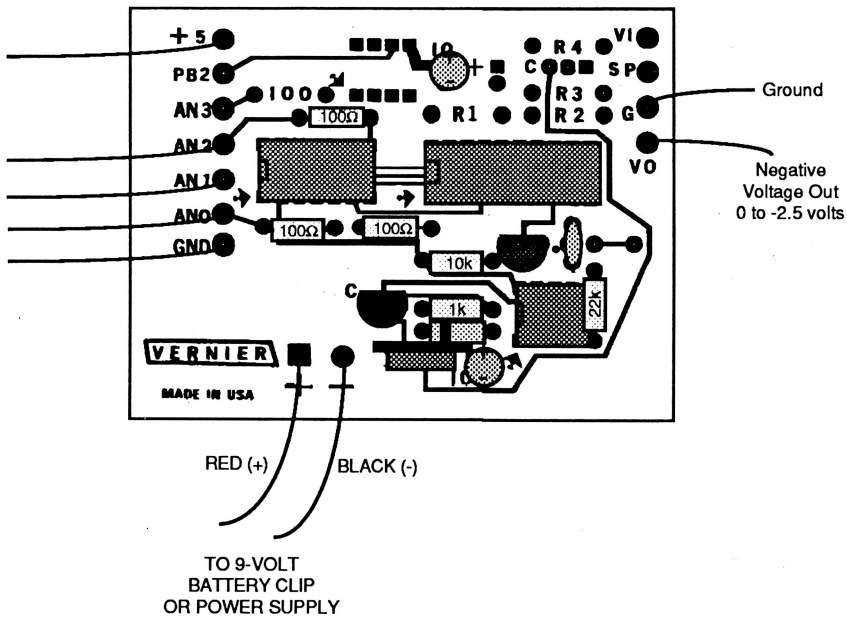


The fine wires in the cable are difficult to attach to the board, and it is easy to have a stray strand cause a short. We recommend you twist the strands of each wire together and coat the wire with a thin layer of solder. If you plan to expand this project later, do not cut the AN3 (pin #12) or PB2 (pin #4) leads too short. They are used for one of the project extensions.

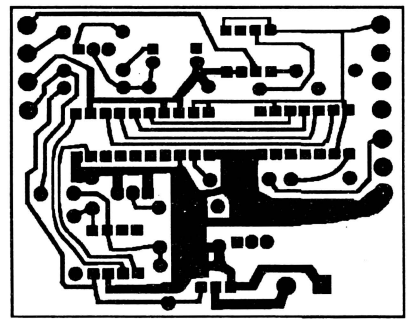
☐ 7. **Connect the wires to the circuit.** Solder these five ribbon cable leads to the circuit board at the corresponding points on the left side of the circuit board.

Solder the battery clip leads to the two holes at the bottom of the circuit board. Usually the "+" lead of a battery clip is red, and the "–" lead is black. If you are unsure, you can use an ohmmeter or a voltmeter to identify the leads.

Solder the leads of the 2-conductor cable to the two holes labeled "G" and V_O on the right edge of the circuit board. These leads are for the output voltage. Label the two wires so you will know which is which when you connect devices to the D-to-A Converter.



☐ 8. **Inspect the circuit board.** Double-check the front side of the circuit board. Make sure all of the components are in the right places and oriented correctly. Take a good look at the back of the circuit board. Make sure all of the leads are soldered. Use wire cutters to trim excess leads. Check for solder bridges between locations that should not be connected. The figure below shows what the back of the circuit board looked like before you started soldering. It will help you spot solder bridges.



The Back Side of the Circuit Board

☐ 9. **Plug in the three ICs.** Plug the three ICs into their sockets. Pin #1 on each of these ICs is marked with a dot or a line (or both) at one end. In all three cases this end goes toward the left. The 74164 goes into the left socket, the DAC0832 into the largest socket, and the LF353 into the smallest socket. Be sure all pins are lined up with the holes and gently push them into their sockets. Make sure the pins don't go outside the hole, or get curled up underneath.

☐ 10. **Test your circuit.** To test the output of the D-to-A Converter, first use a voltmeter set to read DC volts.[3] Plug a 9-volt battery into the battery clip.[4] Plug the 16-pin connector into the Apple game port, place the *Chaos in the Laboratory Programs #2* disk in the disk drive, and turn on the computer. If it does not start up normally, immediately turn it off, disconnect your circuit and refer to the "Troubleshooting" section below. If everything seems to start up correctly, run the program D.TO.A. This simple program lets you enter a number that controls the voltage output. The range of numbers is 0 to 255, which covers the voltage range 0 to -2.5 volts. Entering 0 should produce 0 volts; entering 255 should produce -2.5 volts. The number entered should be approximately the voltage in hundreds of a volt. The program also displays a binary representation of the number entered. You can enter 256 to cause the program to cycle the D-to-A Converter slowly through its range of voltages.

☐ 11. **Use your circuit to control lamps, LEDs, small motors, etc.** The D-to-A Converter can be used to control a wide variety of electrical devices; however, keep the following limitations in mind when you make connections: (1) The voltage range is 0 to -2.5 volts; and (2) Without the addition of extra circuitry, the output should not exceed 160 mA.[5]

Example of things you might try to connect:

- LEDs (include a 100-Ω resistor in series with the LED and connect the cathode (flat side) of the LED to the V_O lead.
- Small electric motors such as the ones sold by Radio Shack (Radio Shack #273-223)
- Small lamps of the type used in flashlights (e.g., Radio Shack #272-1132, #272-1121)

To make sure you are not drawing too much current from the A-to-D circuit, touch the 2709 transistor as soon as you turn on the circuit. It may get warm, but if it is too hot to hold on to, turn off the power and change the external circuit.

When not using the D-to-A Converter, disconnect the battery. The circuit will draw a small current even when you are not using it, so if you leave the battery in place, it will probably end up dead. You may even want to add a switch to the battery circuit.

---

[3]The output of the D-to-A converter goes negative, so if you have a meter that reads only positive voltages, connect the negative lead to the V_O lead and the positive lead to the G lead. If your voltmeter reads both positive and negative voltages (this includes most digital voltmeters), you can connect the leads either way.

[4]Nickle-cadmium batteries designed for use in 9-volt applications can be used here. We recommend them.

[5]One of the project extensions explains how to add to the circuit to allow you to control several amps.

TROUBLESHOOTING

If the D-to-A Converter is not working properly, check for these most common errors:

◊ Are all the leads soldered to the circuit board?
◊ Are all the ICs inserted properly—with pin #1 to the left? Are all the pins seated properly?
◊ Have you made the connections from the 16-pin cable correctly?
◊ Is the 16-pin connector completely and correctly inserted into the game port socket?
◊ Is your 9-volt battery charged? Check its voltage while it is connected to the circuit. If the battery is low, the circuit can act very strangely.

If none of these errors are found, it is time to do some troubleshooting using a voltmeter. Connect the circuit to the computer and connect the 9-volt battery. Set the voltmeter to read DC voltages between -5 volts and +5 volts and connect the negative lead of the voltmeter to the ground output wire. Run the program D.TO.A and set the voltage output to 255. Use the voltmeter to check the voltage at the following points:

DAC SECTION:

IC	Pin #	Expected Voltage	Comments
DAC	19 & 20	5 V	This is the power supply lead from the game port. If it is not close to +5 volts, check your power and ground connections to the game port.
74164	14,2,9	5 V	Same as above.
LF353	8	5 V	Same as above.
DAC	8	2.5 V	This is the reference voltage. It should remain at exactly +2.5 volts. If it does not, make sure the 10-kΩ resistor is connected to the middle lead of the LM385 voltage reference. If all the wiring is ok, the LM385 was probably damaged and must be replaced.
DAC	4-7,13-16	5 V	These lines represent the 8 bits of the byte output from the computer. If this byte is 255, then all 8 lines should be on (near +5 V). If they are not +5 V, then the problem is in the 74164 IC or its connections. If these lines are all +5 V, the problem is in the DAC IC or its connections. Check the control line (AN3) from the computer as well as all the other connections to the DAC IC.
LM7905	Middle	-9 V	This is the power supply from the battery. If this is not at least as low as -7 volts, you need a new battery. If it is 0 or the wrong polarity, then check your connections and the orientation of the LM7905.
LM7905	Right	-5 V	If this is not -5 volts, recheck the wiring of the LM7905 and its orientation.
DAC	11	-2.5 V	This is the direct output from the digital-to-analog converter before it is amplified. If the output is set to 255, then this should be -2.5 volts. If this is correct, then use the program to change the voltage output. You should be able to see this voltage change. If this is working, the trouble is somewhere after this point, so skip ahead to the BUFFER SECTION. If this voltage is not correct, then the problem is in the first half of the circuit.

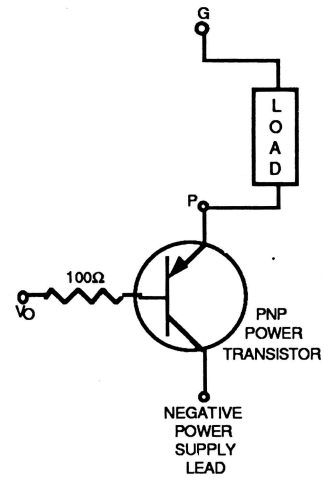
BUFFER SECTION:
Conduct these tests if pin #11 of the DAC is doing its job, but you are not getting an output voltage.

IC	Pin #	Expected Voltage	Comments
LF353	4	-5 V	This is the output of the LM7905 voltage regulator.
LF353	1	-2.5 V	This is the analog output voltage after only one loop through the op-amp. It should read -2.5 V, and you should be able to change it from the computer as you did with the voltage at pin #11 on the DAC. If not, check all connections to the LF353.
LF353	7	-2.5 V	Same as above, but after two passes through the op-amp. Check the connection of the 22-kΩ resistor and the 0.01-μF capacitor, as well as all connections to the LF353.
2907	Middle	-2.5 V	Same voltage as above. If not, check the connections.
2907	Left	-5 V	This is the 7905 voltage regulator output.
2907	Right	-2.5 V	This is the final voltage out. If this is correct, then check the connection to your output wire. If not, check the wiring of the transistor, diode, and 1-kΩ resistor.

Project Extensions

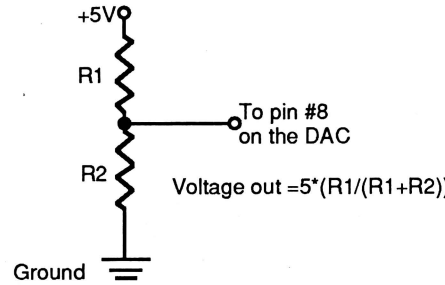
Circuit Modifications

◊ Add a power transistor so that the circuit can supply more current. The schematic diagram at the right shows how to add a Radio Shack transistor (Radio Shack #276-2027 or #276-2043) to provide several amperes of current. When using this circuit, connect the emitter of the transistor directly to the negative of a power supply, not the –5-volt connection on the circuit board. Otherwise, the 7905 voltage regulator which supplies the –5-volt power may be overheated if you use large currents. If you build this project extension, you will probably want to use a power supply that can supply a negative voltage with respect to ground, instead of just a 9-volt battery.



A Power Transistor Circuit

◊ Change the voltage range of the D-to-A Converter. The output voltage will be between 0 and the negative of the reference voltage connected to pin #8 of the DAC0832. In the core project, the reference voltage is supplied by a 2.5-volt regulator. The number of voltage steps always remains 256, so if the range is smaller, the steps get smaller and the precision of the voltage control improves. To use a different reference voltage you will have to disconnect the LM385. If you connect pin #8 of the DAC to the +5-volt power supply line, the output range of the DAC will be 0 to -5 volts. By using two resistors as shown in the diagram below you can change to any reference voltage from 0 to 5 volts. The voltage will be 5*(R2/(R1+R2)). You can also use a potentiometer in place of the two resistors to make a variable voltage. The sum of R1 and R2 must be at least 10 kΩ. If you use a potentiometer, the fixed value resistor must be at least 10 kΩ.



Voltage Divider Circuit for the Reference Voltage

◊ Change the circuit so the output voltages are positive. If the reference voltage applied to pin #8 of the DAC0832 is negative, the output from the op-amps will be positive. You will have to make four other circuit changes: (1) Change the transistor to a NPN transistor; (2) Reverse the orientation of the diode; (3) Cut the trace connecting the –5-volt power from the collector of the op-amp; and (4) Connect a +5-volt external power supply to the collector of the NPN transistor.

Program Improvements

The D.TO.A program is a very short demonstration program. You may want to add features to it. Some ideas are:

◊ Add a display of the voltage output. The program now simply displays the count in decimal and binary format. Calculate and display the voltage.

Experiment with feedback. One example would be to set up the D-to-A Converter to power a small motor as you monitor the speed of the motor. Mount a disk with a hole in it on the motor. Connect a photogate (Project 1 in *How to Build a Better Mousetrap*) to the computer so that the computer can monitor the frequency of the motor's rotation as the hole in the disk goes by. Modify the D.TO.A program so that the computer can monitor the speed of the motor and adjust the output voltage until the motor reaches just the speed you specify. Other feedback examples include temperature-controlled environments and light-level control.

WAVEFORM.MAKER program is a short BASIC program that generates files that can be played back by the SAMPLE.PLAY program. It can be modified to create different waveforms, or to change the frequency or amplitude of the waveform.

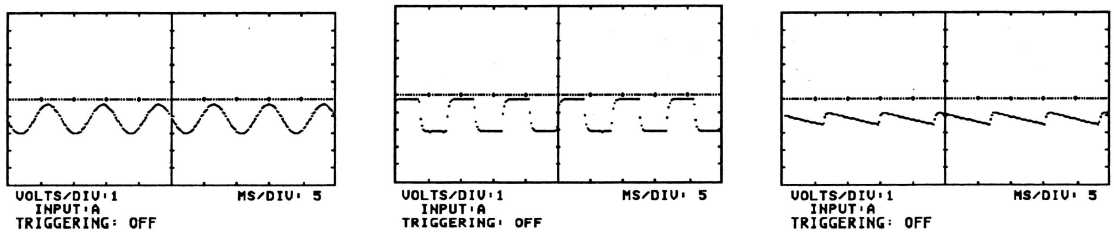
The SAMPLE.PLAY program is more difficult to modify for two reasons: (1) Some of it is written in machine language; and (2) If you add lines to this program, you will have less memory left for data storage. As originally written, the program had room for 129 blocks of data, each one of which is 256 bytes long. This is a total of about 33,000 bytes of data. Using Applesoft BASIC there are only about 36,000 bytes available for the program, variables, and the digitized sound data. Normally data is taken at about 6000 bytes/second, so that allows room for about 6 seconds of sound. If you increase the size of the program, you will have to lower the limit of the number of blocks of memory that can be digitized.

Additional Experiments

A FUNCTION GENERATOR

The Digital-to-Analog Converter can change its output very rapidly—up to 6000 times a second. As a result, it can operate as a function generator to produce an output signal that varies in a controlled way. Three wave shapes are stored on the *Chaos in the Laboratory Programs* disks: sine waves, square waves, and sawtooth waves. Function generators that produce voltage output waveforms like this are often used in laboratories. To see the output of the circuit, you might want to connect the V_O

and G output leads to an oscilloscope. Run the program SAMPLE.PLAY. This program is on the back of the *Chaos in the Laboratory Programs #1* disk. Its volume name is /S1. You should see a list of nine options. Select L - LOAD SOUND. You will then be asked which sound file you wish to load. Type SINE and press <RETURN>. The file should start loading. Press "P" for PLAY SOUND. You should see a sine wave pattern on your oscilloscope. It will be centered at -1.25 volts. Try loading and playing these other waveforms: SQUARE and SAWTOOTH. The graphs below show the output waveforms. Notice that the voltage goes negative in these waveforms.



Analog-to-Digital Converter Playing SINE, SQUARE, and SAWTOOTH Waveforms

Mode A of the SAMPLE.PLAY program alters the delay used in generating a wave. The normal value is 1. Entering a number higher than 1, up to 255, will lower the frequency of the output. Entering a negative number causes the computer to skip samples as it sends out data. Entering -1 is normal, -2 makes the frequency twice as high, etc.

The program WAVEFORM.MAKER is a short BASIC program that was used to generate the files SINE, SQUARE, and SAWTOOTH. You can modify this program to produce other waves.

Of course, one application of a function generator is to connect it to a speaker and make sound. That leads us to the next group of project extensions.

DIGITIZED SOUND

The output of the Analog-to-Digital Converter can be used for a lot of interesting (and entertaining) experiments involving sound. You will need a speaker system. Two different approaches are explained below. The first one is the easiest, since the Vernier Software D-to-A Converter circuit board has the circuitry built in. All you have to do is add the components.

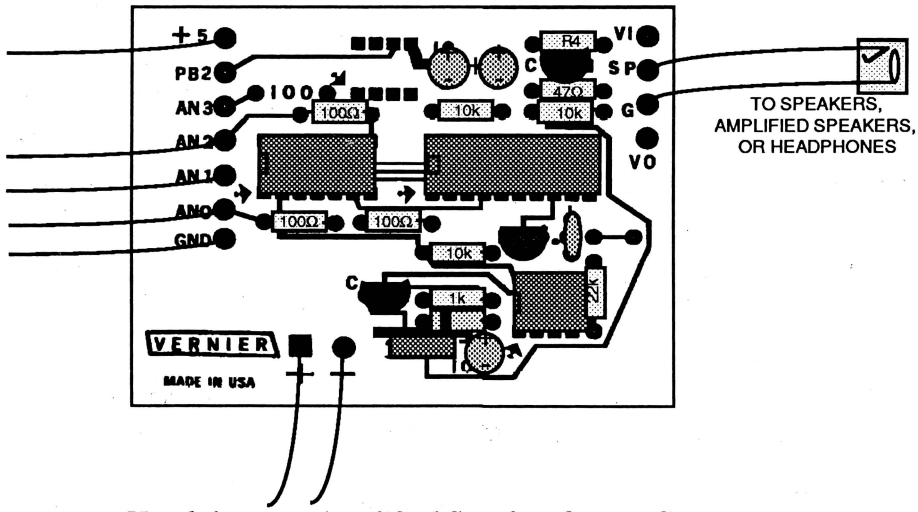
◊ Add a headphone/amplified speaker connector to the circuit.

To play sounds using the digital-to-analog converter and a set of headphones, all you need to do is add a few components to the circuit board. The necessary connections are built into the upper-right corner of the board. You will need to add the following:

- ☐ (1) 10-ΩF capacitor
- ☐ (2) 10-k resistors
- ☐ (1) 47-Ω resistor
- ☐ (1) 2907 transistor
- ☐ (1) 2-conductor, 1/8 inch, in-line phone jack (half of a Radio Shack #274-333)
- ☐ (1) one other resistor, see text below

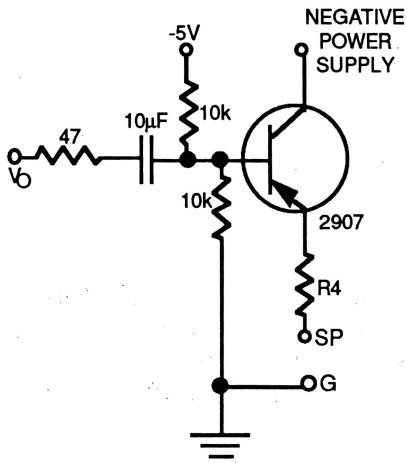
The placement of these components is shown below. The 10-k resistors go in the places labeled "R1" and "R2" on the board. The capacitor goes just to the right of the other 10-μF capacitor. The 47-Ω resistor goes at the spot labeled "R3". The resistor to be used in the position labeled "R4" depends on what you will be connecting to the circuit. If you are using a pair of headphones, or an unamplified speaker, use a small resistance value—33 Ω is about right. If you are using amplified speakers, you will need a much larger value here—1 kΩ is a good starting point. The resistor used

at R4 controls the volume. You may even want to put a variable resistor in this location. If you later mount the whole project in a box, you can put the variable resistor on the box as a volume control.



Headphone or Amplified Speaker Output Connections

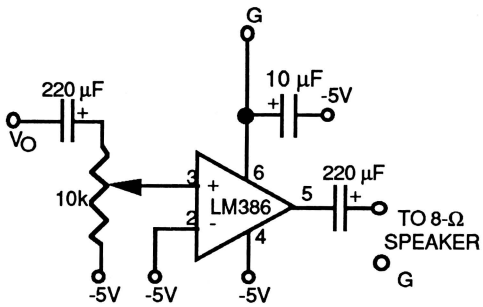
The schematic diagram of the speaker circuit is shown below:



The Speaker Circuit Schematic

◊ Add an amplifier circuit for driving a speaker (optional).

If you do not have amplified speakers and you want to have loud sound, you can add this circuit. It uses an audio amplifier IC (Radio Shack #276-1731) and will produce quite a lot of sound out of any 8-Ω speaker.



An Audio Amplifier Circuit

◊ Try out the Digital-to-Analog Converter as a sound source.

Once you have a speaker or headphones connected, run the program SAMPLE.PLAY. This program is on the back of the *Chaos in the Laboratory Programs #1* disk. Its volume name is /S1. You should see a list of nine options. Select L - LOAD SOUND FROM DISK. You will then be asked which sound file you wish to load. Type TEST and press <RETURN>. The file should start loading. Enter P for PLAY SOUND. You should hear a report on how your circuit is working. Press <RETURN> to stop the sound and return to the menu. If you do not hear anything, go back and check your speaker circuit carefully. Also, make sure your battery is connected and charged.

Enter C for a list of other sounds you can load into memory. We have filled the backs of both *Chaos in the Laboratory Programs* disks with sounds. The table below lists the names of some of the ones you may want to try. Only the back of the #1 disk has the SAMPLE.PLAY program and the ProDOS operating system, so you should start the computer using this disk. The program allows you to specify a file to load by referring to its complete ProDOS pathname; you can actually load files on any disk connected to the computer if you know the volume name of the disk and the file.

Some other options allow you to change the way a sound is played. Option B plays the sound backwards. Option A alters the delay used when playing a sound. The normal value is 1. Entering a number higher than 1, up to 255, will slow down the sound. Entering a negative number causes the computer to skip samples as it sends data to the speaker. Entering -1 is normal, -2 makes it play twice as fast, etc.

Using the Sound Files

The sound files fill the backs of both of the *Chaos in the Laboratory Programs* disks. The back of disk number #1 has the volume label (prefix) /S1. The back of disk #2 has the volume label /S2. If you have two disk drives, place both /S1 and /S2 in the drives and you can load files from either disk by entering the whole pathname. For example the file TRAIN is on the /S2 disk. Even if you are running the SAMPLE.PLAY program from the /S1 disk, you can load the train sound by using /S2/TRAIN as the filename. Here is a list of the original sound files and where they are located:

Sound File Name	Volume	Length in 512 Byte Blocks
TEST	/S1	51
SINE	/S1	11
SQUARE	/S1	11
SAWTOOTH	/S1	11
INDIA.VOYAGE	/S1	66
CLOCK	/S1	51
BEETHOVEN	/S2	16
MR.ED	/S2	51
HONK.HONK	/S2	11
TRAIN	/S2	66
CAR	/S2	66
HICCUP	/S2	9
TARZAN	/S2	16
ILLOGICAL	/S2	9
HIDDEN.MESSAGE	/S2	19

If you have a 3.5-inch disk drive, you can copy all of the sound files to one disk and still have lots of room left over. That makes loading and saving the files easier. Also, if you have a RAM disk, you can load a number of sound files onto it so that they can be accessed very quickly as you run the SAMPLE.PLAY program.

This system normally plays back sounds at about 6000 samples/second. CD players use 44,000 samples/second, so do not expect comparable sound. Low frequency sounds such as talking is played back well. Higher frequencies cannot be as accurately sampled and sound somewhat distorted. Also, this D-to-A converter has 8-bit resolution (256 voltage steps). Audio quality D-to-A converters often have 14 or more bits of resolution (more than 16000 voltage steps).

If you continue with the project extensions, you can start to record your own sound files. Notice that the sound files are very large. You will need to format your own blank disks to make room for saving sound files.

◊ Digitize your own sounds

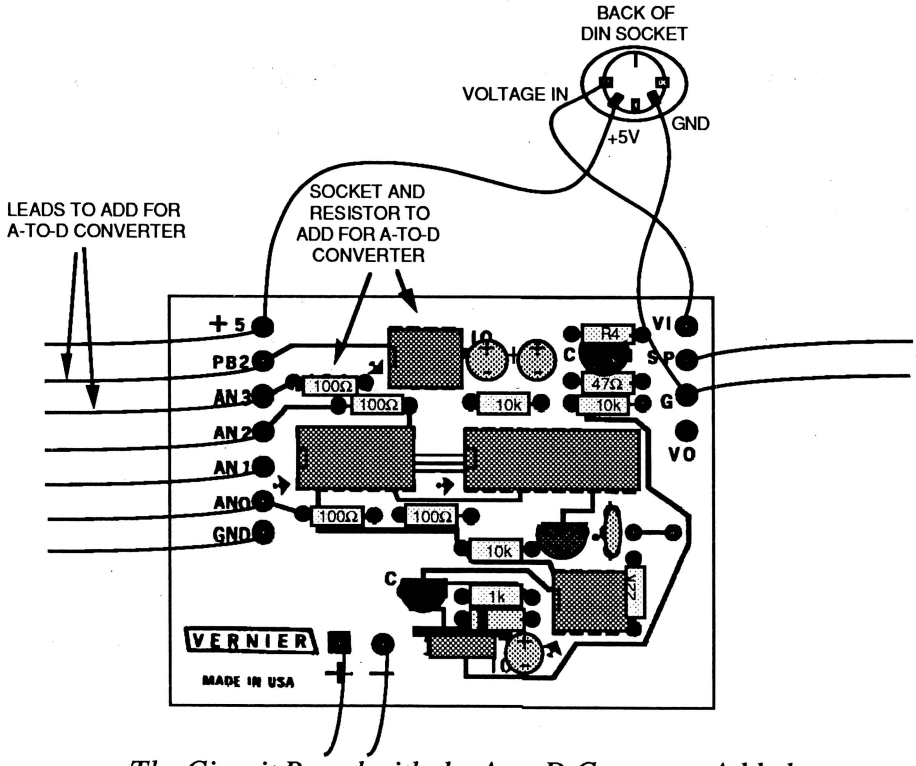
The next step in digitized sound experiments is to add an A-to-D converter to the circuit so that you can digitize your own sounds. The circuit for Project 5 in this book should be used. There are two approaches you can take:

- If you have already built the A-to-D Converter from Project 5, you can just plug it in whenever you want to digitize sounds and then plug in the D-to-A Converter (this project) whenever you want to play them back.
- You can also build the A-to-D Converter right on the D-to-A custom circuit board. In fact, the top left corner of the circuit board has all the wiring already on it. All you have to do is add the components.

The parts you need to add for the A-to-D Converter are:

☐ (1) LTC1092 ADC
☐ (1) 8-pin DIP socket
☐ (1) 100-Ω resistors
☐ (1) 5-pin DIN socket

The component placement and the connection for the input voltage lead are shown in the diagram below. Note that the GND and +5V lines to the DIN connector need to be connected to holes that are already used on the circuit board.



The Circuit Board with the A-to-D Converter Added

You will also need a microphone circuit. The one described as a project extension for Project 5 works very well. In fact, that is the microphone used to record many of the sample sounds on the *Chaos in the Laboratory Programs* disks. The DIN plug on this microphone will plug into the DIN socket on either the A-to-D Converter or into one added to the D-to-A Converter circuit board.

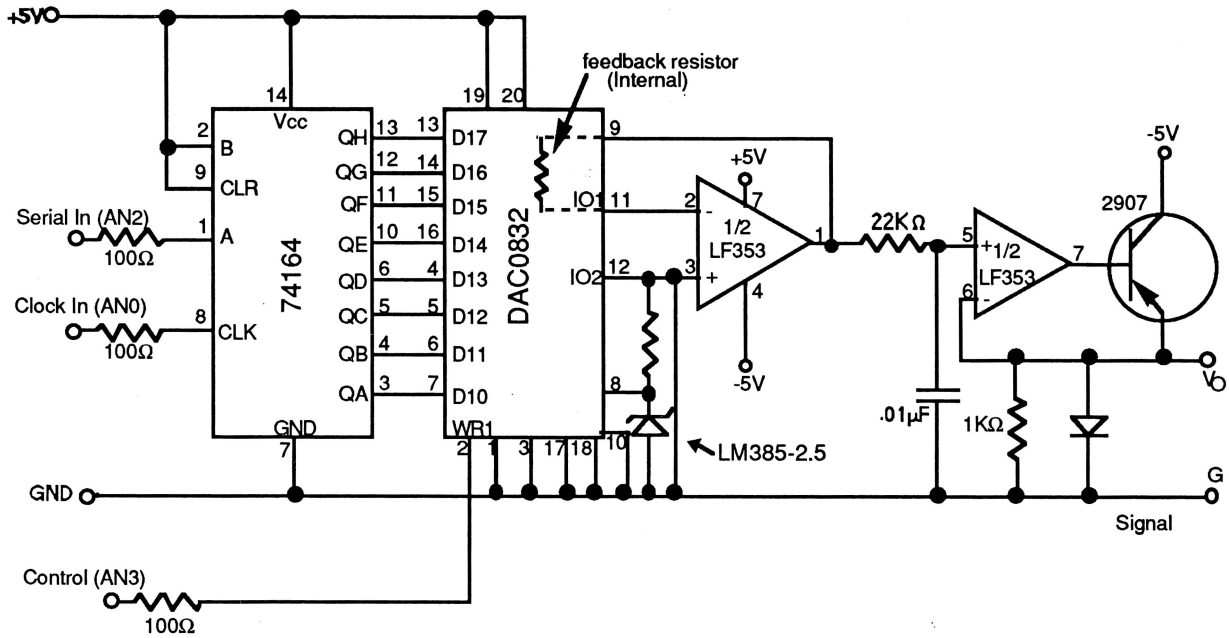
Once you have your microphone and a speaker connected, run the program SAMPLE.PLAY. You should see a list of nine options. Press D for DIGITIZE SOUND. The program will ask how much memory to use. Enter a number from 1 to 129. The default answer is 129. This controls how much data the program will record; 129 blocks take about six seconds.[6] As soon as you press <RETURN> the computer will begin sampling. Get your sound ready, and press <RETURN>. When the sampling is completed, the menu will be displayed. Press P for PLAY SOUND. You should hear what you just digitized. If not, go back and test both your input and output circuits individually. The best way to test the microphone is to use the SCOPE program on the *Chaos in the Laboratory Programs #2* disk.

The other option on the menu that is related to digitizing is Option Z. It allows you to enter a delay value for the digitizing. The delay is normally 1 (as fast as possible—about 6000 samples/second). Entering a higher value will allow you to record longer sounds, but the quality will decrease. Slower sampling will work fairly well for voice, but not very well for music. If you change this delay, you will also have to change the play delay (using Option A to get the sound to play at normal speed. Option F allows you to save your sounds on a disk. You can use just a filename and have the sound saved on the current disk, but there is not much extra space. You can use another disk by specifying the ProDOS prefix before the filename such as /SOUNDS/TEST.SOUND. You can also view the catalog of a disk with Option C. Enter the ProDOS prefix, or just press <RETURN> for the current disk.

Supplemental Information

A More Detailed Look at the Circuit

This section is a fairly detailed description of how the circuit works. As you read through this section, you may want to refer to the schematic diagram of the circuit.



The Schematic Diagram of the D-to-A Converter

[6]The number entered is the number of 256 byte segments to be stored.

When the program sets a voltage, it sends an 8-bit count to the circuit via the game port's AN2 digital output line. The circuit uses a **74164 serial in shift register** to read in these bits one at a time until all 8 are specified. The AN2 game port line is used for the data. The AN0 line from the game port acts as a clock line indicating when each new bit is being sent.

The 8 data lines from the shift register are connected to the input of the **DAC0832 8-bit D-to-A converter**. The count that was sent bit-by-bit to the serial shift register is now represented as 8 signals into the input of the D-to-A converter. The DAC0832 D-to-A converter is designed to work with one **op-amp** to produce an output voltage that varies from 0 volts to the negative of the reference voltage applied to pin #8 of the DAC0832. This voltage is developed on pin #1 of the first op-amp.

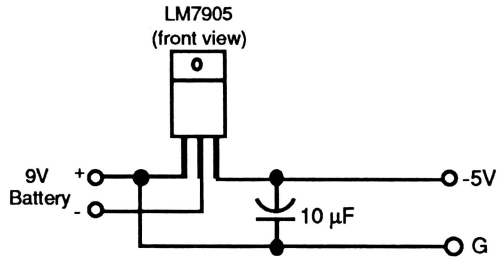
The **LM385Z regulator** provides a steady +2.5-volt reference voltage, which is not influenced by power supply fluctuations.

The **22-kΩ resistor** and the **0.01-μF capacitor**, connected to the output from the first half of the op-amp, act as a low-pass filter. Without this filter, the transition between voltage levels produces very short, high-frequency spikes. The filter removes these spikes and smooths the transition between voltage levels. The filter greatly improves the sound quality when using the D-to-A Converter with speakers.

The filtered voltage signal is next sent to the **second op-amp,** which along with a **2907 transistor**, act as a buffer. These components do not change the voltage, but they allow more current to be supplied at the same voltage. The voltage at the emitter of the transistor (V_O) is the same as the input voltage to the second op-amp, but by using the transistor, the circuit can supply up to 160 mA.

The **diode** connected between the two output terminals prevents voltage spikes produced by inductive loads, such as motors, from damaging the transistor or other circuit components.

The negative voltages needed for this circuit must be supplied by an external power supply. We recommend the following simple circuit using an **LM7905 voltage regulator**.



The Negative Power Supply Circuit

This circuit is built into the D-to-A custom circuit board and it supplies a regulated -5 volts for use by the ICs. Notice that the Digital-to-Analog Converter is designed to use this negative power supply for powering external circuits as well.

You may wonder why we set up this circuit to supply negative voltages. Granted, it would be more convenient to have positive output. If the reference voltage connected to pin #8 of the digital-to-analog converter is negative, the output voltage from the circuit will be positive. So why do it the way we have? The digital-to-analog converter IC requires a –5-volt power supply for operation; therefore, we had to supply a negative voltage with a battery or other external power supply. If the output voltage is made positive, a second external power supply would be required to supply the current for the external load. You cannot use the +5V line on the Apple II to power loads. By making the output of the circuit go negative, we can use one external power supply for both tasks. One of the project extensions explains how to modify the circuit for producing positive voltages if using two external power supplies is not a problem for you.

Program Design Notes

The SAMPLE.PLAY program uses the machine-language program SP.0 for digitizing and playing sounds. You can control the machine-language program easily with BASIC commands.

Waveforms are stored in memory starting at memory location 5120 ($1400 in hexadecimal). The number of 256-byte pages in the waveform is stored in location 5119 ($13FF). This value is limited to 129 pages, which allows for 129*256 = 33024 bytes. This takes all the memory from $1400 to $9600. At the maximum sampling rate, it takes about 6 seconds to fill this memory.

The machine-language routine SP.0 is loaded into memory at 4608 ($1200).

To **play** a waveform, you use the following command in BASIC:

```
CALL 4608, PAGES, DLY, SKIP
```

PAGES gives the length of the waveform. DLY is the delay which slows down the output. SKIP controls whether each value is output, or whether some should be skipped to make the output faster.

To **digitize** a waveform, you use a BASIC command like this:

```
CALL 4614, PAGES, RD
```

PAGES gives the length of the waveform. RD is a delay which controls the rate of sampling.

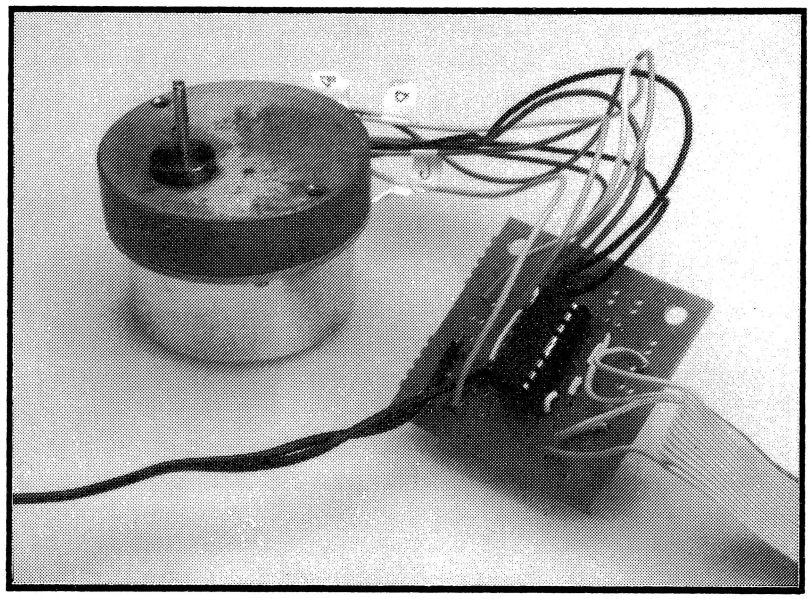
To **play** a waveform **backwards**, you use the following command in BASIC:

```
CALL 4611, PAGES, DLY, SKIP
```

The files containing waveforms (sounds) are structured as follows: The first byte is the number of 256-byte pages that the waveform contains. The rest of the file is the 8-bit analog voltage values taken at even intervals, typically 6000/second. It is fairly easy to create or modify these data files. Take a look at the WAVEFORM.MAKER program as an example. When a waveform file is saved on a disk, a command like this is used:

```
PRINT D$ "BSAVE FILENAME, A$5119, L$"; PAGES*256
```

One problem you may run into when you modify SAMPLE.PLAY is running out of memory. If you add too many lines to the program, you will eventually get errors caused by your BASIC program conflicting with the machine-language routine loaded at $1200.



Project 11

Stepper Motors: Unipolar and Bipolar

by J. Roger Bacon

This project involves controlling *stepper motors* with the computer. A stepper motor is a special kind of motor whose rotor can be precisely controlled. Specialized integrated circuits are available that can generate the proper sequence of electrical signals to operate stepper motors. These ICs will be used in this project. They have a fairly high current rating, so many types of stepper motors can be used without optical isolators, power transistors, or other interfacing circuitry. Stepper motors can be used on a solar tracker, a mobile robot, or any number of projects where objects must be moved under computer control. Two or three stepper motors can be controlled by one computer using this project.

Project Requirements

Computer

This project uses the annunciator output lines, which are available only on the 16-pin game port of the Apple II+, IIe, or IIGS. This project cannot be done using other models of Apple II computers.

Background Knowledge

- Electronics: Experience with soldering components to circuit boards is recommended.
- Programming: None required for the core project. Some experience with BASIC might be helpful for some of the project extensions.

Parts

There are two different versions of this project, depending on whether you are using a unipolar or a bipolar stepper motor.

If you are using a **unipolar**[1] stepper motor (5 or more wires on the motor), you will need the following parts:

- ☐ unipolar (5, 6, or 8 wires) stepper motor with 0.5 amp or lower current rating (All Electronics, R&D Electronics, and other surplus electronics suppliers often have inexpensive unipolar stepper motors. This project will be much easier if the leads of this motor are identified for you. The Vernier Software Unipolar Stepper Motor Parts Kit includes a motor with labeled leads.)
- ☐ Sprague UCN-5804B Stepper Motor Driver IC (These chips are available from Newark Electronics, Hamilton-Avnet Electronics, or Arrow Electronics. You will need one chip for each stepper motor you use.)
- ☐ 16-pin DIP socket (One socket is needed for each stepper motor.)
- ☐ circuit board (These plans assume you are using one-half of a Radio Shack #276-159. There are two circuit boards per pack and one board is required for each stepper motor.)
- ☐ (5) short jumper wires for connections on the circuit board
- ☐ 2-conductor cable (1 m) for connection to the external power supply
- ☐ 16-pin ribbon cable (with a DIP plug on one end) One cable is sufficient for all of your stepper motors, assuming that you will be controlling all the motors with one Apple II computer.

If you are using a **bipolar**[2] stepper motor (4 wires on the motor) you will need the following parts:

- ☐ bipolar (4 wire) stepper motor with 0.35 amp or lower ratings (All Electronics, R&D Electronics, and other surplus electronics suppliers sometimes have inexpensive stepper motors.)
- ☐ Motorola MC3479P Stepper Motor Driver IC (These chips are available from Newark Electronics, Hamilton-Avnet Electronics or Arrow Electronics. One IC is required for each stepper motor you plan to use.)
- ☐ 16-pin DIP socket (One socket is needed for each stepper motor.)
- ☐ circuit board (These plans assume that you are using one half of a Radio Shack #276-159. There are two circuit boards per pack and one board is required for each stepper motor.)
- ☐ 27-kΩ carbon resistor, 1/4 watt (one for each stepper motor) (red, violet, orange, gold)
- ☐ (3) short jumper wires for connections on the circuit board
- ☐ 2-conductor cable (1 m) for connection to the external power supply
- ☐ 16-pin ribbon cable (with a DIP plug on one end) One cable is sufficient for all of your stepper motors, assuming that you will be controlling all the motors with one Apple II computer.

Note: All of the parts for the unipolar stepper motor project are available from Vernier Software in a parts kits. The cost of the Unipolar Stepper Motor Parts Kit is $16.00 (Order Code CSU-16).

Tools and Supplies

The most important supply for this project is a good DC power supply. Stepper motors require quite a lot of current. If you are using a unipolar stepper motor, the power supply should be capable of supplying up to 3 amps at 5 volts. If you are using a bipolar stepper motor, it should be capable of supplying up to about one amp at 12 volts. If you plan to use more than one motor, the power supply will need to supply even more current. Batteries can be used for this project, but they will be drained fairly quickly. Other tools and supplies needed are: soldering iron, solder, wire strippers, and an ohmmeter.

---

[1]Unipolar stepper motors are also called *four-phase* or *bifilar* stepper motors.

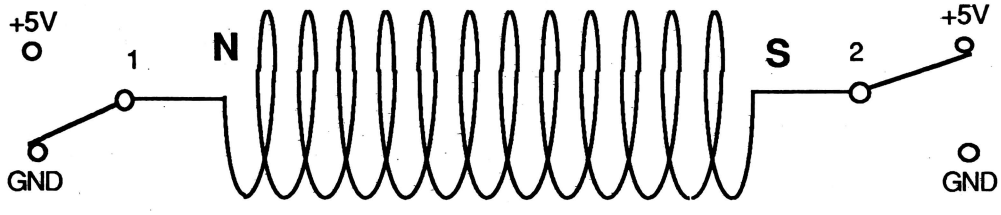
[2]Bipolar stepper motors are also called *two-phase* or *unifilar* stepper motors.

Background Information

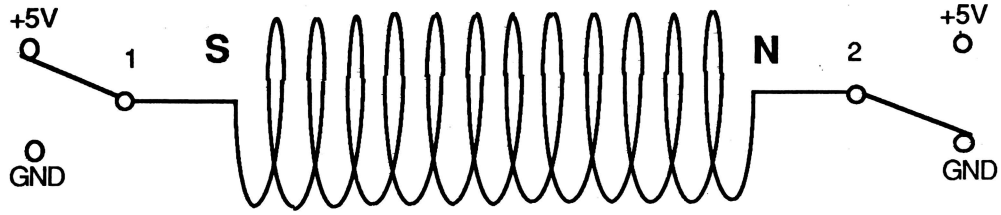
Stepper motors are precise motors which advance one increment each time the electrical signals to their lead wires are changed. They are used in printers to position the print head in the correct spot. You hear a stepper motor in action when you use a floppy disk drive. It moves the read/write head of the disk drive to different tracks of the disk. The series of signals necessary to advance the motor can be generated by a computer program.[3] Specialized stepper motor driver integrated circuits are also available which take care of the generation of the correct pattern of signals. Many of these ICs have a fairly high current rating, so the stepper motors can be used without optical isolators or power transistors. That is what we will be doing in this project.

Stepper motors, like all other electrical motors, operate because an electrical current creates a magnetic field which is either attracted or repelled by another magnet. In this case, the second magnet is the rotor. By changing the direction of the current in the coils surrounding the rotor, the rotor turns due to the magnetic forces of the changing current. In a stepper motor, the direction of the current in the coils surrounding the rotor is controlled in a pattern that forces the rotor to move one step either clockwise or counterclockwise.

There are two types of stepper motors, *bipolar* and *unipolar*. A bipolar stepper motor has coils wired such that they can switch polarity. The two diagrams below show how the polarity of the magnetic field can be changed by changing the position of two switches.



Magnetic Field with Current Directed to the Left



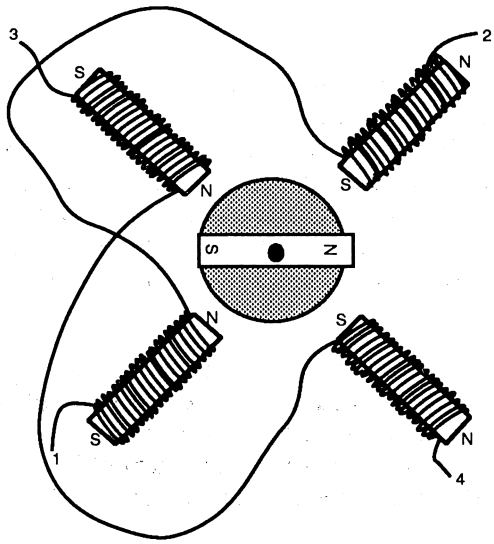
Magnetic Field with Current Directed to the Right

In each bipolar stepper motor, there are many sets of four such electromagnets (typically 10-60 sets). We will try to explain the operation of a stepper motor by simplifying how a bipolar stepper motor works. If we wanted to construct a stepper motor that had only four positions (4 steps in a complete rotation), we would need only one set of four electromagnets.[4] The four are coupled together in two pairs as shown below. In a stable position, the rotor and the four electromagnets would look something like the diagram on the left. By switching the polarity of one of the coupled sets of magnets, we can get the rotor to move as shown in the diagram on the right.

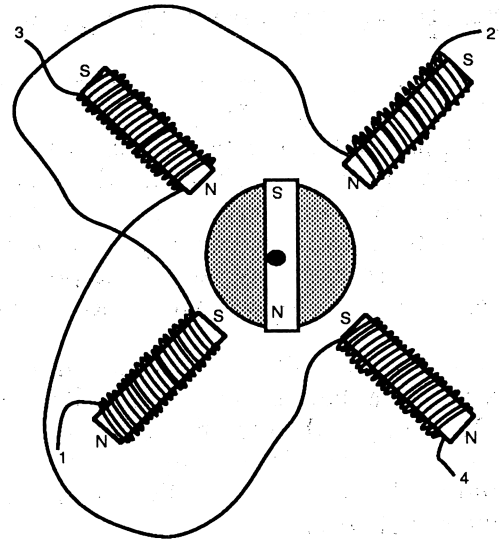
---

[3]The book *How to Build A Better Mousetrap* includes a project that does just that. The computer controls four optically isolated switches which turn on and off the coils of a unipolar stepper motor in the proper sequence.

[4]Actually, it is possible to have eight positions using one set of four magnets by incrementing by half-steps. This process is outlined in the "Supplemental Information" section.



Four-Position Stepper Motor

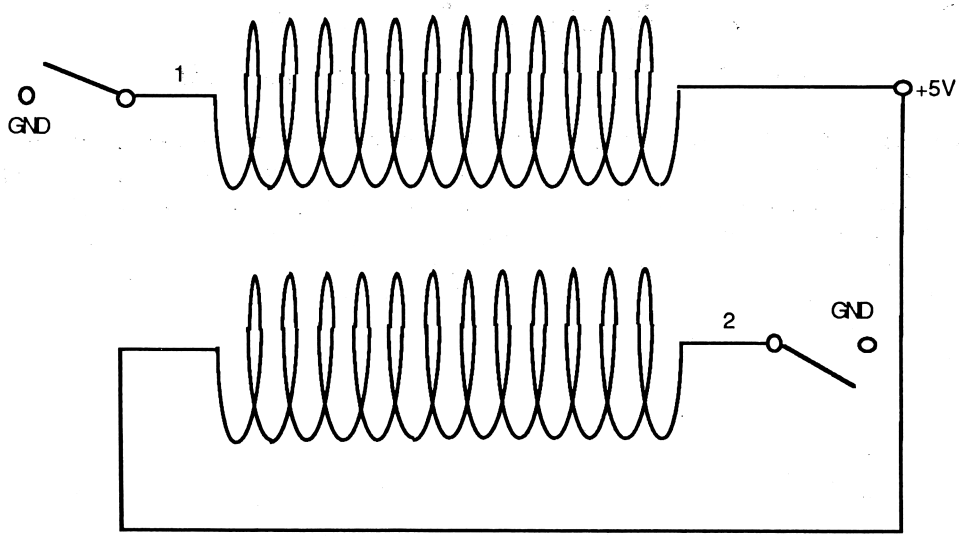


Four-Position Stepper Motor After One Step Clockwise

Each movement is called a *step,* and each step requires a different pattern of electrical signals. As we can see in the simple example above, we had to switch the polarity of one of the magnet pairs to get the rotor to advance one step clockwise. To advance another step in the same direction, we would have to switch the polarity of the other pair of electromagnets. By working our way through a pattern of settings, we can make the rotor move in a complete rotation.

A real bipolar stepper motor works in a similar way. All bipolar stepper motors have four lead wires. One pair of leads connects to a coil of wire that controls the magnetic field at a number of different locations around the rotor. Our simple example had one pair of wires controlling the polarity of just two locations. More electromagnets around the circumference mean that each step is fairly small. Typically, there are 48 to 200 steps in a complete rotation.

In a unipolar stepper motor, each electromagnet has two windings on it. A particular winding can have only one polarity (that is where the name *unipolar* comes from). One of the two windings is activated to set up a magnetic field. The two windings are wrapped and wired so that the polarity of the electromagnet is reversed, depending on which one is on. In the figure below, both coils of wire would be wrapped around the same metal core. The polarity of the electromagnet changes depending on which of the two switches is closed.

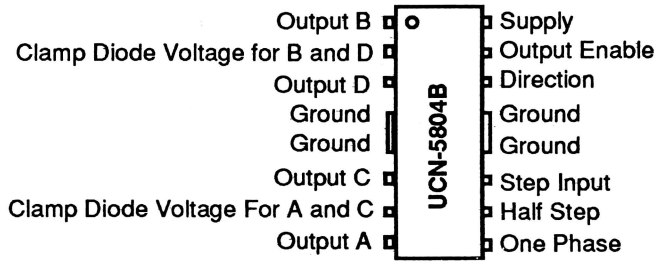


A Unipolar Electromagnet

A complete unipolar stepper motor would have two groups of electromagnets of the type diagrammed above. Note that there are two leads coming from the coils, plus the positive power supply lead. This means a unipolar stepper motor will have at least five lead wires.[5] This is the easiest way to tell unipolar and bipolar stepper motors apart, since bipolar stepper motors always have four lead wires.

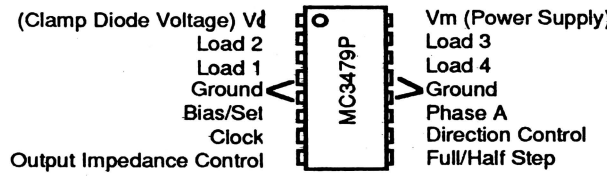
Activating a particular pattern of coils causes the motor rotor to move (attraction and repulsion again). By switching one or the other of the switches to ground, we can get current to flow in either direction, creating a magnetic field in either direction. In reality, unipolar coils are not two separate coils, but they are wrapped upon the same core.

We will use two different stepper motor driver ICs. The Sprague UCN-5804B is a 16-pin driver for unipolar stepper motors. It has a fairly high current capacity (1.5 amps at up to 50 volts).



Pinout for the Sprague UCN-5804B

The Motorola MC3479P is a 16-pin driver chip for bipolar stepper motors with a 16.5 volt, 350 milliamp/coil limitation. Notice that the current limits of this IC are a little lower than those of the Sprague unipolar driver IC.



Pinout for the Motorola MC3479P

Many small stepper motors will operate directly from these chips without the need for power-boosting circuits. The control part of these driver chips operates at TTL logic levels. This means that the annunciator output lines of the Apple II game port can be used to send the control signal to the stepper motor driver IC. For either of these ICs, two inputs are required. One input controls the direction of rotation of the motor. If this line is at 5 volts, the motor rotates clockwise. If this line is at 0 volts, the motor rotates counterclockwise. The other control line controls the stepping. Each time the output line controlling the step pulses (low->high), the motor advances one step.

One nice thing about this project is that the Apple II game port has four annunciator output lines and only two are needed to control a stepper motor using one of these ICs. You can easily control two stepper motors. In fact, if you connect the same output line to the "direction control" line of three motors, you can even control three stepper motors from one computer.

The "Step-by-Step Instructions" below are divided into two sections. The first section is for unipolar stepper motors and the other is for bipolar stepper motors. The unipolar stepper motor project has a higher current rating and therefore can handle larger motors. Both of the projects are controlled using the annunciator output line on the 16-pin Apple II game port. Note that this means that Apple IIc, IIc+ and Laser 128 computers cannot be used for this project.

---

[5]Sometimes extra power supply leads are provided, so that unipolar stepper motors occasionally have 6 or 8 lead wires.

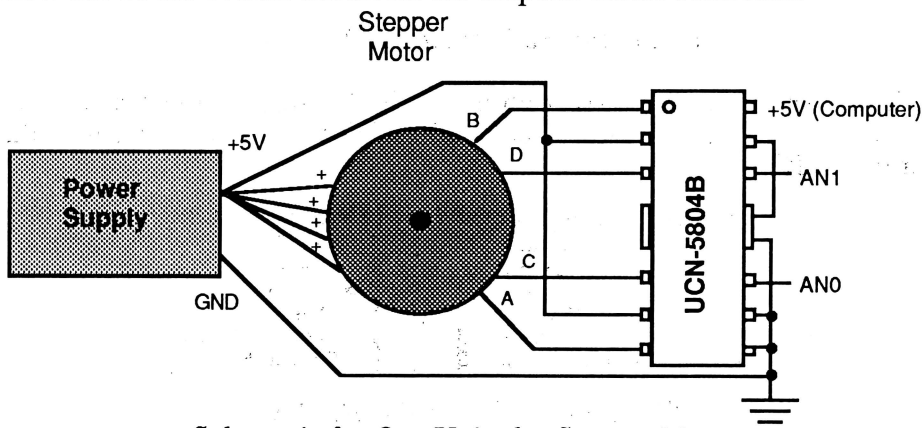
the annunciator output line on the 16-pin Apple II game port. Note that this means that Apple IIc, IIc+ and Laser 128 computers cannot be used for this project.

Stepper motors can be used for many different project extensions. Among those suggested below are a solar tracker, a mobile robot, and automatic window blinds. Stepper motors can also be used for computer-controlled adjustments that involve rotation; for example, the rotation of a robot arm or the changing of the wavelength setting on a instrument. Since each rotation step is the same size and is both reversible and reproducible, it is easy to make precise adjustments. In some projects you may want to operate a stepper motor with two lines and use the other two annunciator lines to control other parts of the project. A stepper motor driven syringe with an electrically-controlled valve is an example.

Step-by-Step Instructions for Unipolar Stepper Motors

Sprague Electronics produces a stepper motor driver IC for unipolar stepper motors, the UCN-5804B. The connections shown below are for one stepper motor, however, two motors can easily be controlled with one computer. For the second motor, a second stepper motor driver chip is used with AN2 and AN3 controlling the number of steps and the direction of rotation, respectively. If you are using two motors, substitute the program TWO.MOTORS for MOTOR.TEST in the instructions below.

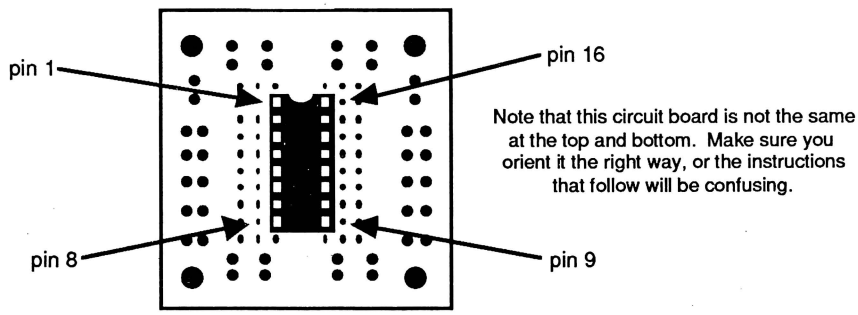
The diagram below shows the overall circuit for the unipolar motor controller.



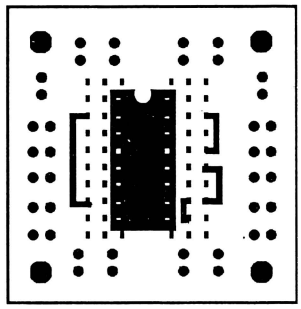
Schematic for One Unipolar Stepper Motor

These instructions assume that you are using one half of a Radio Shack #276-159 circuit board. If you are going to use two motors, leave the two halves of the circuit board together. Otherwise, break the circuit board apart and use just one half. It is also assumed that you are using a Unipolar Stepper Motor Parts Kit from Vernier Software. This kit includes information for identifying the leads of the stepper motor (A, B, C, D and +). If you are using a unipolar stepper motor without identified leads, you will have to figure out the correct leads by trial and error or from data sheets provided with the motor.

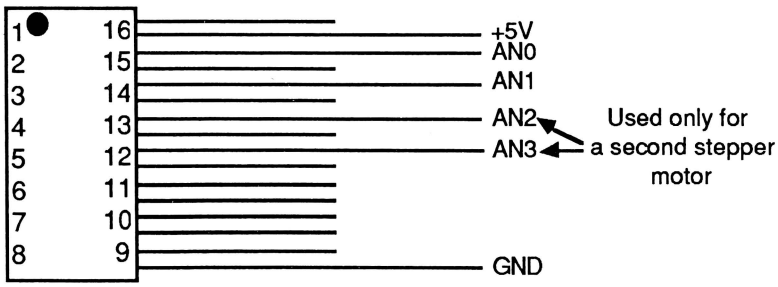
☐ **1. Solder the socket onto the circuit board as illustrated below.** The socket should be inserted from the plain side (no printing or copper plating) with each pin going through to the copper-plated side. Notice the positioning of the notch in the socket. The pins of this socket are referred to by number. The top-left pin in the figure is referred to as pin #1 and the rest are numbered in a counterclockwise pattern around the socket. Solder all 16 pins to the copper-plated side of the circuit board. Do not plug in the IC at this time.



☐ **2. Solder 4 jumper wires for the +5V and ground connections.** For correct operation of the stepper motor IC, pins #9, #10, and #15 should all be connected to ground (pins #12 and #13), as indicated on the right side of the circuit board shown below. A jumper wire from pin #2 to pin #7 should be installed on the left side of the circuit board as indicated in this diagram. The wires should be inserted into the board and bent so they stay in place while soldering. Cut off the excess leads on the plated side of the circuit board after soldering.

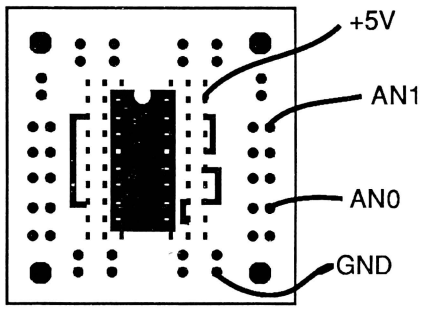


☐ **3. Connect the ribbon cable leads.** A 16-pin ribbon cable should be used to make the connections between the Apple II game port and the circuit board. The +5V and GND leads are used to power the IC. The AN0 line connects to pin #11 (the step input pin) and controls when the motor takes a step. The AN1 line connects to pin #14 and controls the direction of rotation for the motor. Annunciators 2 and 3 are used only if a second stepper motor will be controlled. The diagram below shows the connections to be used on the 16-pin cable.



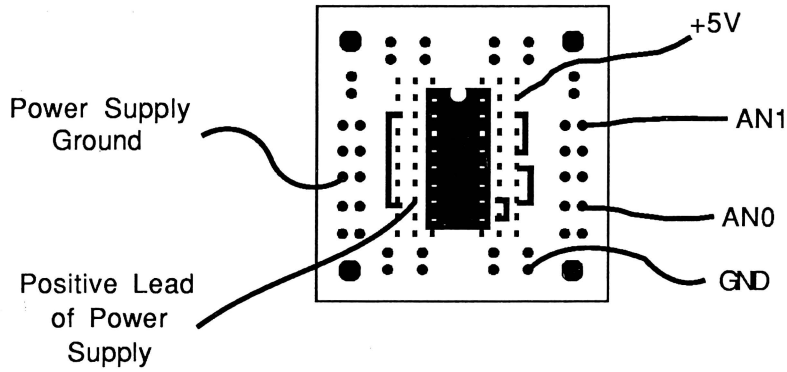
Pin 1: +5V
Pin 8: GND
Pin 15: AN0
Pin 14: AN1
Pin 13: AN2
Pin 12: AN3

The Connections to Use on the 16-Pin Ribbon Cable

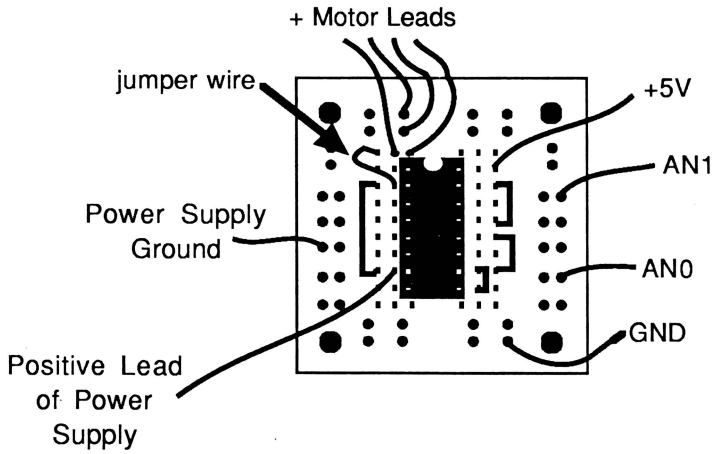


Circuit Board Connections for One Stepper Motor

☐ **4. Connect the power supply wires to the circuit board.** Strip the insulation off both ends of a 2-conductor cable long enough to reach from the circuit board to your external power supply. Solder one lead in the empty hole next to pin #7 on the IC socket. This wire will be connected to the positive terminal of your power supply. Label it "+". The other wire should go into one of the holes connected with pin #5. It will connect to the power supply ground. Label it "Ground". **Note that the +5V from the computer should not be connected to the power supply.**

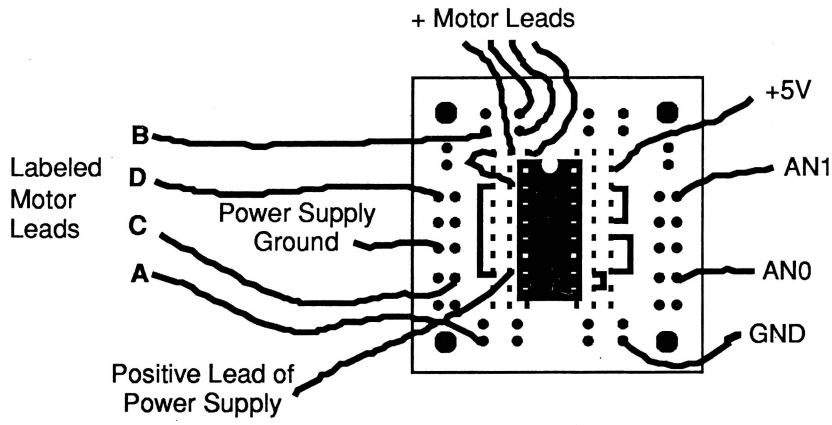


☐ **5. Connect the "+" stepper motor leads to the power supply.** The top set of holes on the left side of the circuit board will be used as the motor voltage supply terminal. Connect all of the stepper motor "+" lead wires to this set of connected holes. Connect a jumper wire from this same set of holes to pin #2 of the socket as shown below. Note: This step assumes that you are using a unipolar stepper motor from Vernier Software. This kit includes a sheet describing how to identify the leads.[6] If you are using a stepper motor from a different source, you may have to experiment to determine the pattern of motor leads either from data sheets or by trial and error.



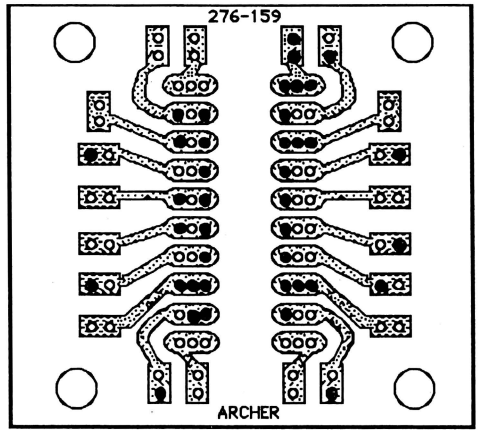
☐ **6. Connect the four other motor leads.** The four other stepper motor leads (A, B, C, and D) should be connected to the holes on the left side of the circuit board. The diagram below shows their connection points.

[6]We do this because we sometimes change the stepper motors we use.



Completed Unipolar Stepper Motor Controller

☐ **7. Double-check the circuit board.** Your solder joints should look smooth and shiny, and there should be no solder bridges between any two points that should not be connected. Cut off any excess wire on the plated side of the circuit board. If you have followed the suggested wiring placements in the previous instructions, then the back of the circuit board should look like this:



Complete Circuit Board as Viewed from the Back

☐ **8. Insert the UCN-5804B chip in the socket.** Make sure that pin #1 (marked with an indentation or a dot) on the chip is positioned at the end of the socket with the notch (the top of the circuit board in all of the diagrams used here).

☐ **9. Connect the wires to the external power supply.** The 2-conductor cable from the circuit board should be connected to your external power supply. The wire pin #7 (labeled "+") goes to the positive power supply terminal. The wire from pin #5 (labeled "–") goes to the negative power supply terminal. **Do not turn on the power supply yet.**

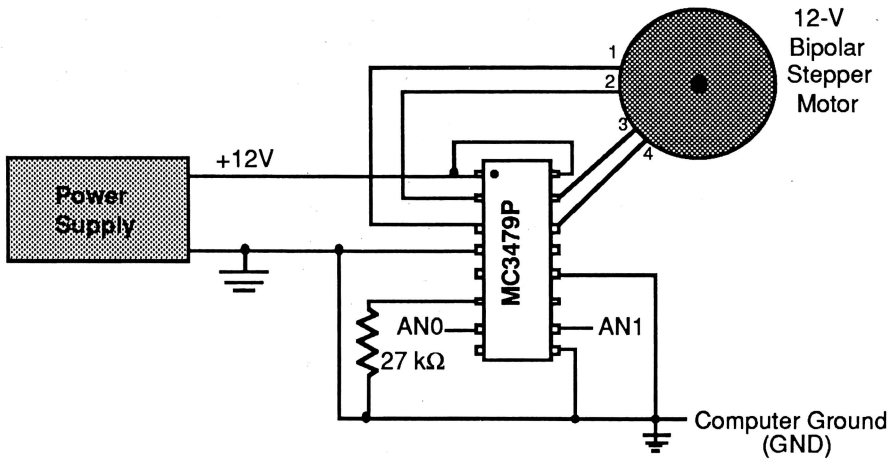
☐ **10. Start up the computer and test the circuit.** Connect the 16-pin ribbon cable to the computer's game port. Make sure you orient the plug correctly in the game port connector. Place the *Chaos in the Laboratory Programs #2* disk in the disk drive and turn on the computer. If the computer does not start up normally, turn it off and check your circuit for a wiring error. Otherwise, run the program MOTOR.TEST. After the program starts, turn on the external power supply and set it to +5V. As soon as that supply is turned on, check the UCN-5804B chip to see if it is getting hot. It may be warm even if the motor is not running, but if it is hot enough to burn your finger, turn off the power supply. You probably have a serious wiring error, and you should refer to the "Troubleshooting" section below.

☐ **11. If the chip seems ok, proceed with the program.** Try out the motor both clockwise and counterclockwise. Note that stepper motors do not move fast, so you may have to watch carefully to see the movement of the shaft. Put a piece of tape on the shaft if you want to make the movement more visible. If the motor wiggles but does not turn, proceed to the "Troubleshooting" section.

☐ **12. (Optional) Connect a second unipolar stepper motor.** If you plan to use two motors, repeat these step-by-step instructions for the second motor. The wiring of the second stepper motor is the same except AN2 replaces AN0 and AN3 replaces AN1. A separate IC and circuit board are required for each motor. The same power supply and ribbon cable can be used. Make sure your power supply can supply enough current for two motors. Use the program TWO.MOTORS if you have connected a second stepper motor.

Step-by-Step Instructions for Bipolar Stepper Motors

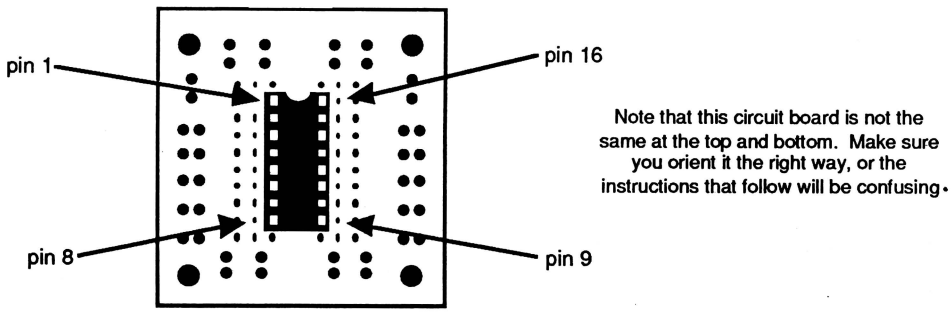
The diagram below shows the overall circuit for a bipolar stepper motor controller using a Motorola MC3479P. Pins #4, #5, #12, and #13 of the IC are connected internally.



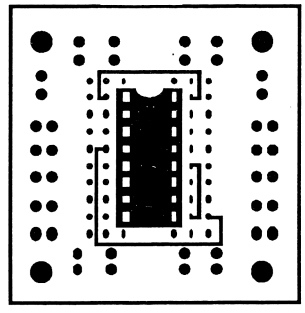
Schematic for one Bipolar Stepper Motor

These instructions assume that you are using one-half of a Radio Shack #276-159 circuit board. If you are going to use two motors, leave the two halves of the circuit board together. Otherwise, break the circuit board apart and use just one half. This project also requires a 12-volt DC power supply.

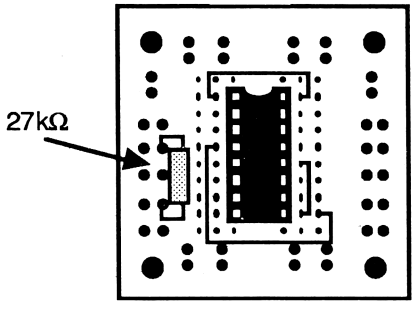
☐ **1. Solder the socket onto the circuit board as illustrated below.** The socket should be inserted from the plain side (no printing or copper plating) with each pin going through to the copper-plated side. The notch in the socket goes toward the top of the circuit board. The pins of this socket are referred to by number. The top-left pin in the figure is referred to as pin #1 and the rest are numbered in a counterclockwise pattern around the socket. Solder all 16 pins of the socket to the copper-plated side of the circuit board.



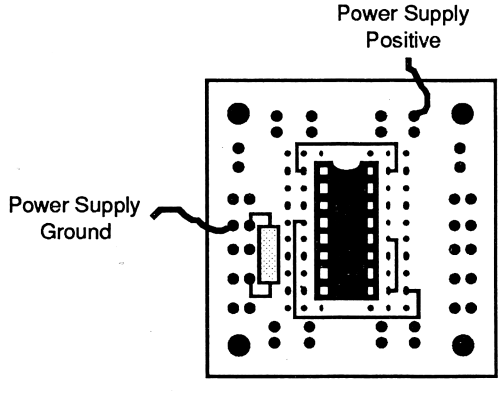
☐ **2. Connect the 3 jumper wires.** Solder jumper wires from pin #9 to pin #12 and also from pin #4 to pin #9. Connect pins #1 and #16 using a third jumper wire. Cut off the excess lead wire on the plated side of the circuit board. The circuit board with all the jumper wires is shown here.



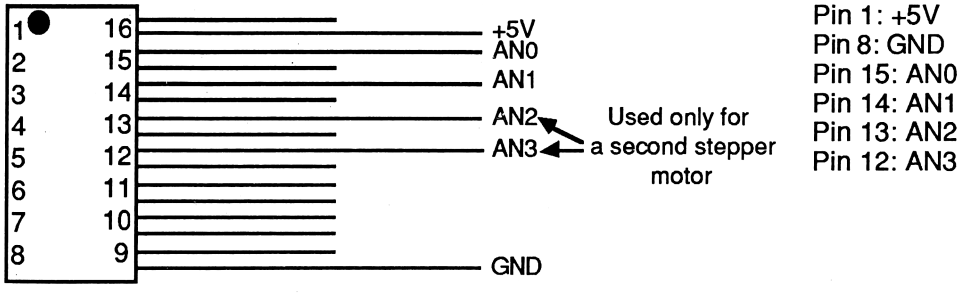
☐ **3. Add the resistor to the circuit.** Connect the 27-kΩ resistor between pins #4 and #6. The bands on the resistor should be red, violet, orange, gold.



☐ **4. Connect the power supply wires to the circuit board.** Strip the insulation off of both ends of a 2-conductor cable long enough to reach from the circuit board to your external power supply. Solder one lead to the hole near the top of the circuit board as shown below. This wire will be connected to the positive terminal of your power supply. Label it "+". The other wire should go into one of the of the holes connected with pin #4. It will connect to the power supply ground. Label it "Ground".

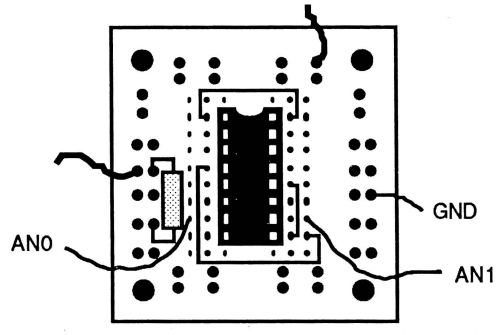


☐ **5. Connect the ribbon cable.** A 16-pin header cable should be used to make the connections between the computer game port and the circuit board. The AN0 line connects to pin #7 (the step input pin) and controls the number of steps the motor advances. The AN1 line connects to pin #10 and controls the direction of rotation for the motor. The AN2 and AN3 annunciator lines are used only if a second stepper motor is to be controlled. The GND lead is connected to pin #4.



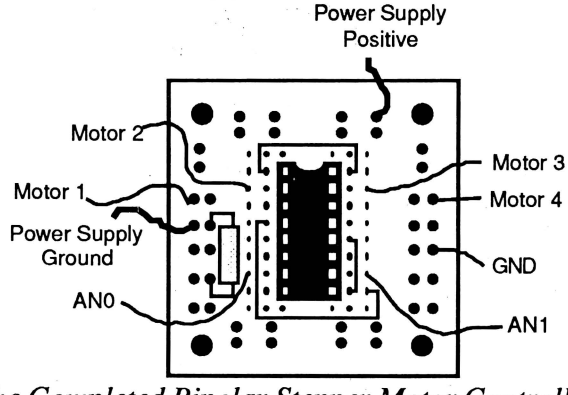
The Connections to Use on the 16-Pin Ribbon Cable

Solder the two annunciator lines and the GND lead to the circuit board at the locations shown below.



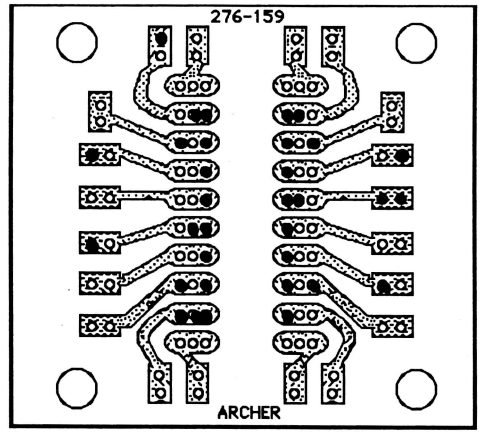
The Completed Bipolar Stepper Motor Controller

☐ **6. Connect the motor leads.** You will first need to identify which leads connect to the same coil, either by experimentation or from information provided by the manufacturer. We refer to the leads by number with wire 1 and wire 2 connecting to one coil and wire 3 and wire 4 connecting to the other coil.[7] Solder motor wire 1 to pin #3. Solder motor wire 2 to pin #2, motor wire 3 to pin #15, and motor wire 4 to pin #14 as illustrated in the diagram below.



The Completed Bipolar Stepper Motor Controller

☐ **7. Double-check all of the connections.** The copper-plated side should look like the illustration below. The only holes that should have leads in them are marked. All of the solder connections should be shiny. Make sure there are not solder bridges. Cut off all excess lead wires.



The Back of the Circuit Board, Showing All Connections

[7] Here is a procedure for identifying the leads using an ohmmeter: Pick one of the leads and label it #1. Connect one of your meter leads to this wire and then test the resistance to each of the other three lead wires. In one case, the resistance should be only a few ohms. Label the lead with the lowest resistance #2. Label the other two wires #3 and #4. There is still a 50% chance that you have your wires labeled incorrectly. If you complete the motor circuit, test it, and if it just wiggles but does not rotate, then switch wire #1 with wire #3 and wire #2 with wire #4.

☐ **8. Plug in the MC3479P IC.** The IC should be plugged into the socket on the circuit board. Make sure pin #1 of the IC (marked with an indentation or dot) is placed at the notched end of the socket. This is the end of the socket at the top of all the diagrams. Be careful not to bend the pins as you insert the IC.

☐ **9. Connect the circuit to the computer and the power supply.** Plug the ribbon cable into the Apple II 16-pin game port. Make sure you orient the plug correctly. Place the *Chaos in the Laboratory Programs #2* disk in the disk drive and turn on the computer. If the computer does not start up normally, refer to the "Troubleshooting" section below. Now connect the two power supply wires to your external power supply. The power supply wires should be labeled "+" and "Ground". Turn on the power supply and set it to +12 V. Check to see if the IC gets hot. If it does, turn the power supply off. A slight warming of the IC is normal, but it should not be hot enough to burn your finger.

☐ **10. Try out the circuit.** Run the program MOTOR.TEST from the *Chaos in the Laboratory Programs #2* disk to try out the motor. If the motor fails to work correctly, refer to the "Troubleshooting" section below. Note that stepper motors do not move fast, so you may have to watch carefully to see the movement of the shaft. Put a piece of tape on the shaft if you want to make the movement more visible.

☐ **11. (Optional) Connect a second stepper motor.** If you plan to use two motors, the wiring is the same except AN2 replaces AN0 and AN3 replaces AN1. A separate IC and a separate circuit board are required for each motor. Use the program TWO.MOTORS if you have connected a second stepper motor.

Troubleshooting

◊ The most important aspect of troubleshooting these circuits is to avoid letting the IC overheat. In normal operation, the motor and the chip are both warm, even if the motor is not rotating, but they should not be hot enough to burn your finger. If the IC gets very hot, disconnect the power immediately, and double-check the wiring.

◊ First check the most common mistakes: Do you have the leads on the 16-pin ribbon cable connected correctly? Is the IC inserted into the socket with pin #1 in the right position? Is your power supply on and connected properly? Is it supplying the proper voltage (5 volts for a unipolar stepper motor or 12 volts for a bipolar stepper motor)?

◊ If the IC gets hot and the motor fails to turn, you may have a missing or bad ground connection. Check to be sure the computer ground, the power supply ground, and the IC ground are all connected. A multimeter in the ohmmeter mode can be used to check these connections. When doing any resistance measurements, the power supply should be turned off and the circuit disconnected from the computer. There should be near-zero resistance between any two of the connections mentioned above. It is a good idea to wiggle these wires as you make the resistance measurements. This might help locate a bad solder joint.

◊ Check the ribbon cable connections by measuring the resistance from the pins on the 16-pin DIP connector all the way to the appropriate socket hole on the circuit board. Correct any connections that fail to give the expected response. Usually a wiring error or a cold solder joint causes the problem. Reheat any questionable joints with the soldering iron. The other common defect is a solder bridge (any solder connection between two points that should not be connected). A solder wick or a solder sucker can be used to remove excess solder.

◊ Check the leads from the stepper motor. Do they go where they are supposed to? If any wire is connected incorrectly (or not at all), the motor may wiggle but not turn. Correct any mistakes you find.

◊ Check with a voltmeter or logic probe to determine if there are pulse signals at output pins of the IC. These are pins #1, #3, #6, and #8 on a unipolar stepper motor, and pins #2, #3, #14, and #15 on a bipolar stepper motor. Run the MOTOR.TEST program and choose slow rotation for a number of turns. Watch the voltage at these pins. Each of pins should go high (near the power supply voltage) and low (near 0) in a regular pattern. <Control> <C> will stop the program after your tests have been completed. If you have pulses at the output pins indicating the IC is working correctly, the error must be in the motor wiring. Double-check all solder joints and be sure all wires are in their correct positions.

◊ If you did not get pulses in the test above, check the annunciator output lines from the game port. With the MOTOR.TEST program running, check the voltage on the AN0 line. It should go high and low regularly. The AN1 line should change from high to low when you select a different direction of rotation for the motor.

For Unipolar Stepper Motors Only:

◊ Be sure all the jumper wires on the circuit board are connected correctly. Use an ohmmeter to check the resistance between any two of the following pins; #4, #5, #9, #10, #12, #13, and #15. It should be near zero. Check various combinations. The resistance should also be zero between pins #2 and #7. It is a good idea to remove the chip and check the resistance at the socket holes. Again, this helps to check for cold solder joints.

◊ The resistance across the motor coils can also be measured using the ohmmeter. For most unipolar motors it should be about 15 ohms. The resistance from pin #2 to pins #1, #3, #6 or #8 should be approximately 15 ohms. A very different reading may indicate a problem with the motor.

For Bipolar Stepper Motors Only:

◊ Check the motor leads with an ohmmeter. The resistance between lead 1 and lead 2 should be only a few ohms (<200 Ω). Similarly, the resistance between leads 3 and 4 should be low. The resistance between other pairs of leads should be very high.

◊ If the motor wiggles, but does not rotate, switch wires 1 with 3 and 2 with 4.

Project Extensions

Circuit Modifications

◊ Experiment with half-step drive

Using either of these stepper motor controller ICs, you can have the motor function in half-step mode. In this mode, a new position is added between each standard step by activating only one set of coils. The net effect is more accurate positioning, but some loss of power and speed. To change to half-step operation, you need to connect one pin of the IC to the power supply voltage. Using a unipolar motor and a UCN-5804B, connect pin #10 to +5V. Using a bipolar motor and a MC3479P, connect pin #9 to the +12 volt external power supply.

Program Improvements

◊ The MOTOR.TEST program is just the start of a useful program. Add your own instructions to make the stepper motor do what you want it to do. The program TWO.MOTORS is a similar program for controlling two stepper motors. Refer to the "Program Design Notes" as you make your changes to these programs.

◊ If you are an experienced machine-language programmer, you might want to rewrite the MOTOR.TEST program in machine language to control one or more stepper motors. The increased

speed of operations would probably be helpful in some applications. Keep in mind that there is a limit to the speed at which stepper motors can be driven.

Additional Experiments

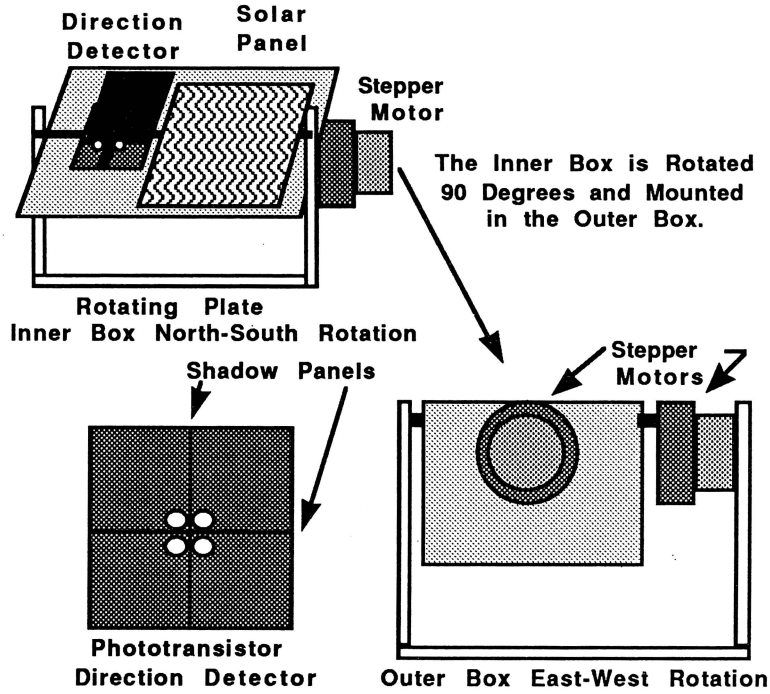
The Stepper Motor can be used for any number of projects in which control of a motion is required. Several examples are listed below, but you are encouraged to think up your own. Try combining the stepper motor with a sensor, so that the motor responds to a change in its environment.

◊ Translate the rotational motion into linear motion

For many applications of stepper motors, you will want to cause motion in a straight line. You can do this in a number of ways. The simplest is probably to use a pulley driven by the stepper motor and string which pulls on the object to be moved. A continuous loop system can be used, and has the advantage of both "pulling and pushing." Make at least one extra loop on the pulley to prevent slippage. Much more elaborate and versatile methods are used in commercial stepper motor applications.

◊ Solar tracker

Build a solar tracker using four phototransistors and a shadow device to sense the position of the sun. One stepper motor can move the solar tracker in the east-west direction and the second would move it in the north-south direction. A light sensor panel can be mounted on the tracker so the intensity of the sunlight will produce a signal that the computer can detect. With the appropriate software, you can have the computer aim a solar collector panel at the sun automatically. This idea is used in many commercial solar power projects. For a detailed explanation of this project, refer to *Apple II Applications* by Marvin L. De Jong.



Solar Tracker Using Stepper Motors

◊ Environmental control

Use stepper motors to raise and lower blinds in a house to control room temperature. Sensors could include photocells (for sunlight) and thermistors (for temperature). The stepper motors may need to be modified so they control motion in a line to move certain types of window blinds. Other similar examples include moving a shade over a greenhouse and opening a vent when a room gets too hot.

◊ Design a motor, or motors, which close a container when it starts to rain. In environmental studies, it is often desirable to collect dry fall (dust, soot, etc.) separately from rain. A stepper motor would allow you to automate this collection.

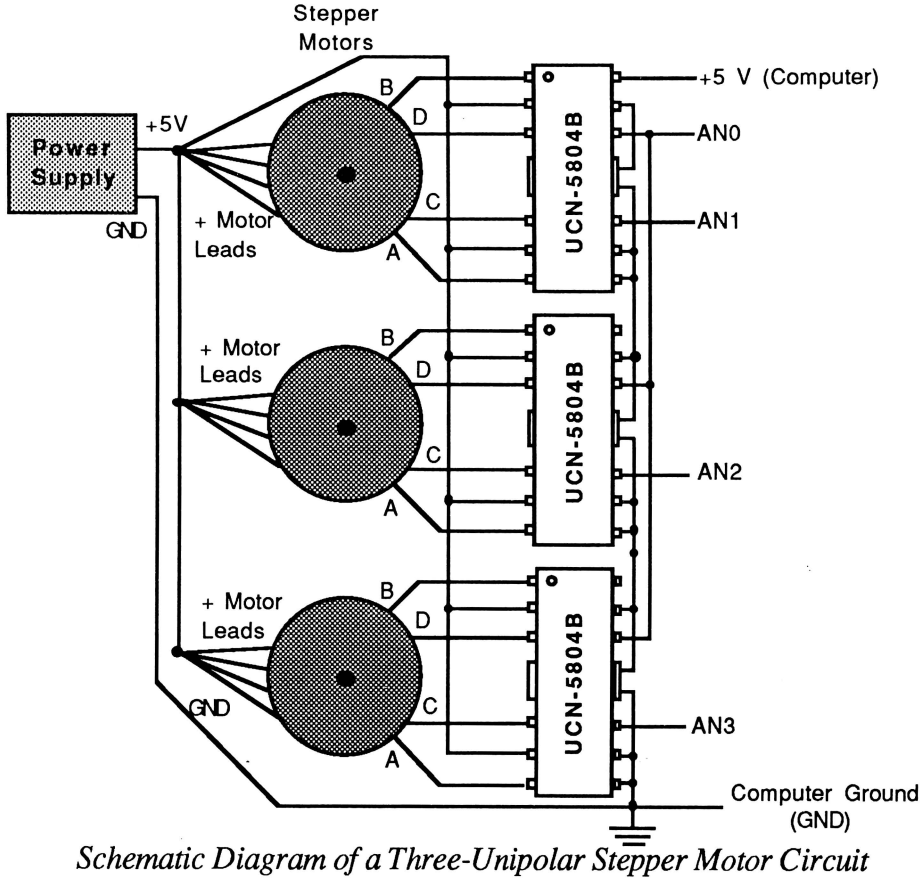
◊ Robotics

Build a two-motor robot with one stepper motor for each drive wheel. The robot could be controlled via the keyboard ("R" for right, "L" for left, "B" for backward, etc.). The device could also be set up to follow a light beam (using a photocell to sense the light intensity).

◊ An Ultrasonic Range Finder (Project 13) can be mounted on a stepper motor so that the Range Finder scans, checking for objects within range. This could be part of an alarm system or a navigation system on a moving robot.

◊ Controlling three stepper motors

There are some projects where two stepper motors would not be enough. Robotic projects, for example, would be very limited with only two modes of movement. Three stepper motors can easily be controlled using one computer, if you are willing to settle for reduced speed. Let's assume that AN0 controls the direction for all three motors, AN1 provides steps to motor 1, AN2 to motor 2, and AN3 the motor 3. To run motor 1 and 2 forward while motor 3 runs backward would require stepping AN1 and AN2 (on and off), then changing AN0, and stepping AN3 (on and off). AN0 would have to be reset to the original condition before the process is repeated. The following diagram using Sprague UCN-5804B ICs and unipolar stepper motors is one possible idea. This project would require some additional programming. Use the TWO.MOTORS program as a starting point.



Schematic Diagram of a Three-Unipolar Stepper Motor Circuit

◊ Controlling up to seven stepper motors

As many as seven stepper motors can be controlled with one computer if a 74LS138 (Three to Eight Line Decoder) is used. Again, AN0 can serve to change the direction on all of the stepper motors, and AN1, AN2, and AN3 can be used to address the decoder chip. Each output from the decoder chip would go to the step input on one of the stepper motor driver ICs. The 74LS138 has eight outputs, but one needs to be left blank to serve as the off cycle (all motors stopped). It will take a lot of programming to get all the correct POKES to make the motors do the desired task. Since each step would require several POKE statements, the motors would run slower. This project would require a power supply with a very high current rating.

Supplemental Information

Program Design Notes

The MOTOR.TEST program is a simple program to demonstrate how a stepper motor can be controlled using the circuits in this chapter. You are encouraged to examine and modify it. The program is outlined below:

Lines 5 to 120: Initialization
This section set variables and constants to the proper values. Key variables used are:
NQ = default number of steps
S$ = speed
S = delay used to control speed
A through H = annunciator control addresses

Line 120: Initializes the annunciators to all off

Lines 130 to 190: Set variables for a new run
For each use of the program, this section allows the user to choose how the motor is to be operated. Variables set are:
N$ = entry of number of steps (positive values are for clockwise rotation, negative values for counterclockwise rotation), "Q" indicates "quit")
DIR = direction of rotation (assigned +1 for clockwise rotation, -1 for counterclockwise rotation)
S$ = rotation speed: Fast (F), Medium (M) or Slow (S)

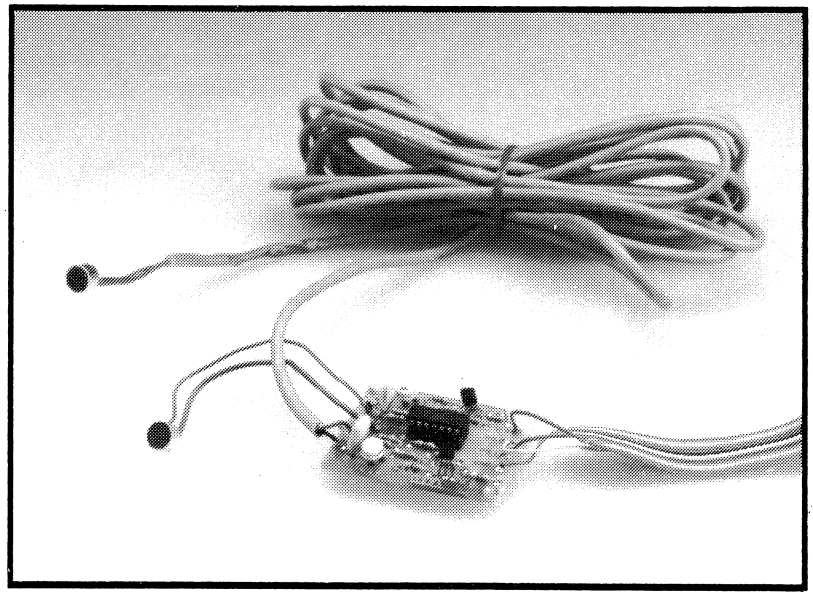
Lines 210-260: Motor operation
This section sends the step signals to the AN0 line at the proper rate and in the number specified by the operator. The AN1 line is set to control the direction of rotation.

Line 350: Sends control back to set variables for a new run

Line 400: Stops the program when Q is entered

The TWO.MOTORS program is similar, but has a few additions to control the second motor. The variables used are:

Q1, Q2 = number of steps requested for motor 1 and motor 2
N1, N2 = number of steps completed for motor 1 and motor 2
D1, D2 = direction of rotation for motors 1 and 2



Project 12

Speed of Sound

This project allows you to measure the speed of sound using the computer, two microphones, and a simple circuit. A sound is produced directly in front of the first microphone, and the computer then times how long it takes for the sound to travel from the first microphone to the second microphone. The computer reports this time, which you can use to calculate the speed of sound.

Project Requirements

Computer

This project can be done with any Apple II computer.

Background Knowledge

- Electronics: If you use the Vernier Software Speed of Sound Parts Kit, which includes a custom printed circuit board, this project is fairly easy. You simply place the components at the proper locations on the circuit board and solder the leads. Experience with soldering is recommended. If you do not use the custom circuit board, the project will be more difficult. In that case, you will need to use just the schematic diagram as your guide to assembly.

- Programming: None required for the core project. Some of the project extensions require experience with BASIC programming.

Parts

- ☐ LM3900 (This IC is a Norton Quad Op Amp. Note that this is a very unusual op-amp—do not try to substitute a different quad op-amp)
- ☐ (2) electret microphone (Radio Shack #270-092)
- ☐ 14-pin socket (Radio Shack #276-1999)
- ☐ (2) 0.1-μF capacitor (Radio Shack #272-1069)
- ☐ 10-μF electrolytic capacitor (Radio Shack #272-1025)
- ☐ (2) 2.2-MΩ, 1/4 watt, metal-film resistor (red, red, black, yellow, brown)
- ☐ (2) 1-MΩ, 1/4 watt, metal-film resistor (brown, black, black, yellow, brown)
- ☐ (4) 10-MΩ, 1/4 watt, carbon resistor (brown, black, blue, gold)
- ☐ (2) 4.7-MΩ, 1/4 watt, carbon resistor (yellow, purple, green, gold)
- ☐ (2) 22-kΩ, 1/4 watt, carbon resistor (red, red, orange, gold)
- ☐ (2) 560-Ω, 1/4 watt, carbon resistor (green, blue, brown, gold)
- ☐ (2) 47 Ω, 1/4 watt, carbon resistor (yellow, purple, black, gold)
- ☐ (2) 100-kΩ potentiometer (Radio Shack #271-284)
- ☐ (2) 2907 PNP transistor (or equivalent)
- ☐ heat-shrink tubing (4 cm) or electrical tape
- ☐ 3 meters or more of 3-conductor cable
- ☐ circuit board (This project is greatly simplified if you use the custom circuit board that is included in the Vernier Software Speed of Sound Parts Kit. If you use a general-purpose circuit board, make sure it is large enough for all of the components.)
- ☐ cable to game port (see *Appendix A*)

Note: A Speed of Sound Parts Kit containing all of the components needed for this project, including a custom-printed circuit board, is available from Vernier Software. This kit sells for $19.00 and is available with either a 16-pin cable for connection to the internal game port (Order Code CSS-16) or a 9-pin cable for connection to the external game port (Order Code CSS-9). An order form is provided at the back of this book.

Tools and Supplies

Soldering iron, solder, wire cutters, wire strippers, and a multimeter (DC voltage and resistance).

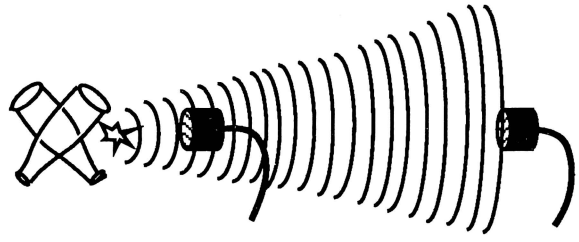
Background Information

Sound waves are mechanical waves moving through air (or any other medium) by an interaction between molecules. The speed of any mechanical wave depends on the medium through which it travels. In general, the velocity depends on the density of the material and its elasticity. For sound waves in air, the speed of sound turns out to be about 343 m/sec in air at room temperature (20°C). The speed depends on the temperature, increasing as the temperature increases. It also changes with air pressure and humidity. In gasses other than air, the speed of sound can be very different. For example, in carbon dioxide the speed is about 260 m/s, and in hydrogen the speed is over 1200 m/s.

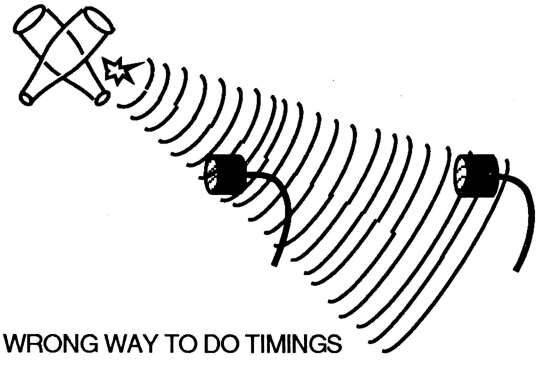
In this project you will build a circuit to measure the speed of sound, using the computer to do the timing. Computers are often used to time things in science labs. In most of these applications, photogates are used to start and stop timings.[1] In this project, you will build two microphone/amplifier circuits that will trigger whenever sound is detected and control the timing. To

[1] *Photogates* are devices that include a light source and a light detector. Whenever an object moves through the photogate, the light beam is blocked and the voltage signal from the light sensor changes. This change can be detected by the computer.

measure the speed of sound, the two electret microphones are placed several meters apart. Then, a sharp, short sound is made near the first microphone and in a direct line with the second microphone. The sound must be produced directly in front of the first microphone and directly in line with the second microphone. If you do not have the sound source and the two microphones lined up properly, the calculation of the speed of sound will be more complicated.[2]



RIGHT WAY TO DO TIMINGS



WRONG WAY TO DO TIMINGS

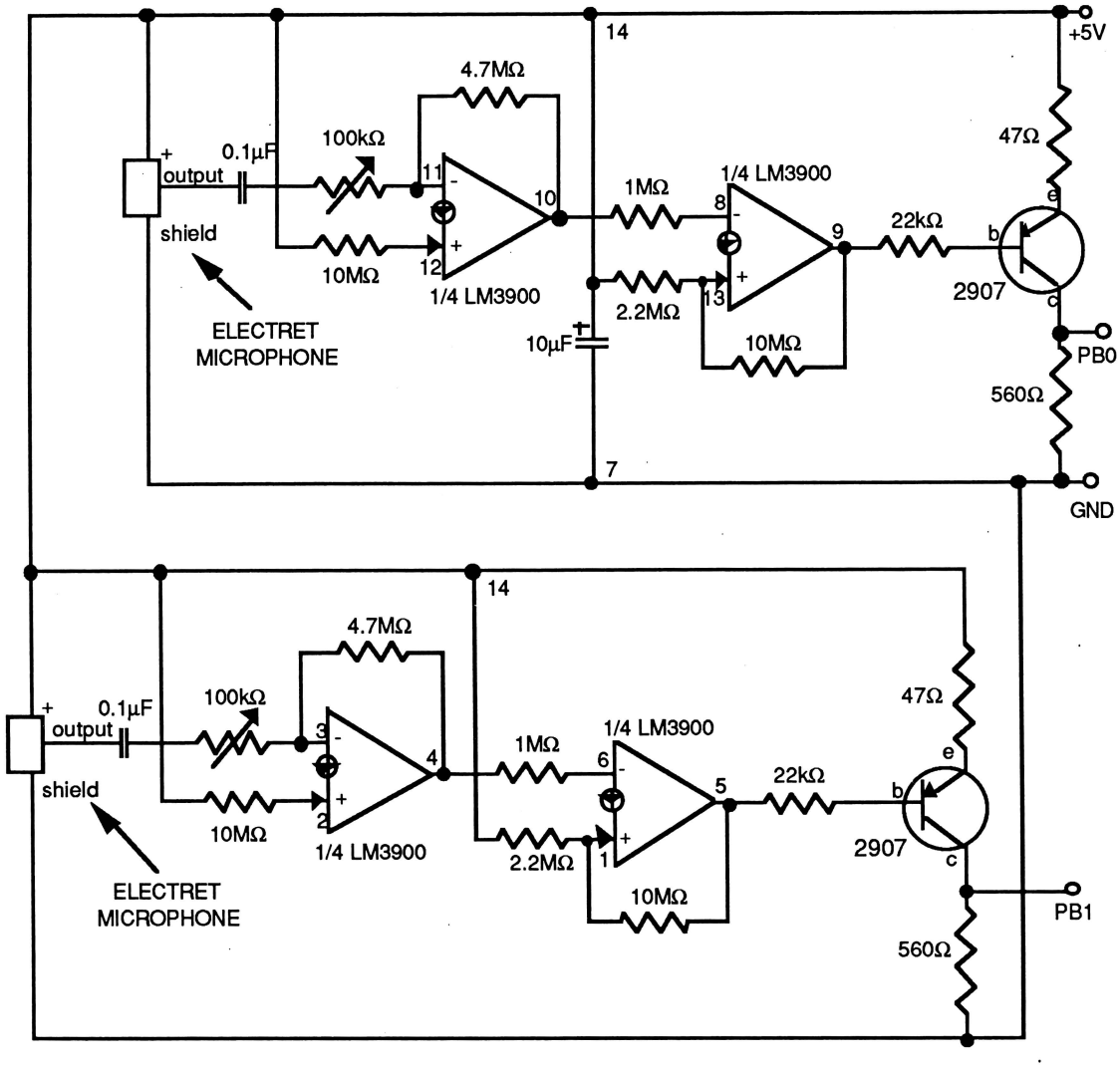
Timing will begin when the first microphone detects the sound. When the second microphone detects the sound, the timing will be stopped. The speed of sound can be calculated from:

$$\text{speed} = \frac{\text{distance between microphones}}{\text{time}}$$

You may be wondering if the timing is affected by the fact that the signal from one of the microphones has to travel down several meters of wire. Doesn't this also cause a delay before the computer receives the stop timing signal? The answer is that it does cause a delay, but it is very small. Electrical signals in a wire travel at nearly the speed of light (3×10^8 m/s). That means that even if you had 10 meters of wire between the computer and the second microphone, the delay caused as the signal travels through the wire is less than a microsecond—too small to be measured by this project.

The complete Speed of Sound circuit is shown below:

[2]The time measured starts when sound first strikes microphone #1 and continues until sound first strikes microphone #2. If the two microphones are in line, this time is the time it takes sound to travel from one microphone to the other. If the microphones are not in line, the time reported represents the time it takes sound to travel the difference in distances from the sound source to the two different microphones.



Speed of Sound Schematic

Here is a brief explanation of how the circuit works. The top and bottom halves of the circuit are essentially the same. Both provide a signal to the computer—the top one to start the timing, the bottom one to stop the timing.

When any kind of sound is detected by the microphone, it produces a current proportional to the intensity of this sound. This signal passes through the 0.1-µF capacitor (which filters out the DC component of the signal) and on to the first op-amp.

The LM3900 op-amp is a current *difference* amplifier. It outputs a voltage proportional to the difference between the two *currents* flowing into its "+" and "–" terminals. This is different from the usual op-amp, which amplifies the difference between the two *voltages* at its terminals. In fact, this current difference amplifying op-amp has a special name, the *Norton* op-amp. In this first stage, the op-amp simply amplifies the current from the microphone, because it compares the microphone current with the constant current flowing through the 10-MΩ resistor. The voltage at the output of the first op-amp varies with the sound received.

The second op-amp is set up to act as a *Schmitt trigger*. It amplifies the signal so much that the output signal is either 5V or ground. Therefore, in effect, we get a "squared-up" version of the input signal.

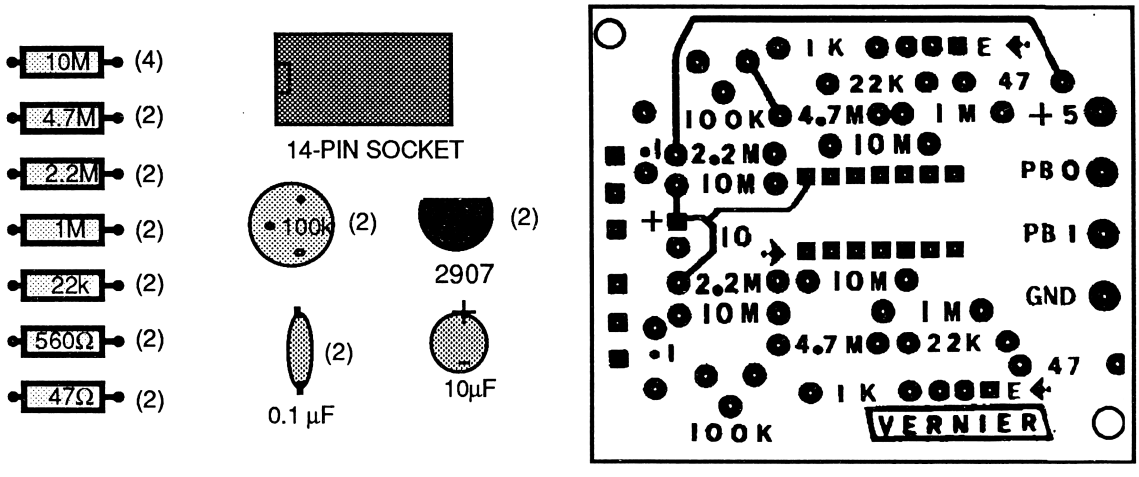
The transistors are included in the circuit to provide enough output current to allow the circuit to be used with all types of Apple II computers. The transistors provide the extra current needed to use the Speed of Sound project on an Apple IIc, IIc+, or Laser.

The potentiometer allows you to set the sensitivity level for the microphone. Before doing timings, check the status of each of the inputs by running the program SOUND on the *Chaos in the Laboratory Programs #1* disk and choosing S - STATUS CHECK. Make sure that each input is LOW before you start the experiment. If one or both of the inputs change from LOW to HIGH when there is no sound, then the potentiometers should be adjusted so that the microphones are less sensitive.

You can use this Speed of Sound circuit for a variety of interesting experiments. Several experiments are suggested in the "Project Extensions" section.

Step-by-Step Instructions

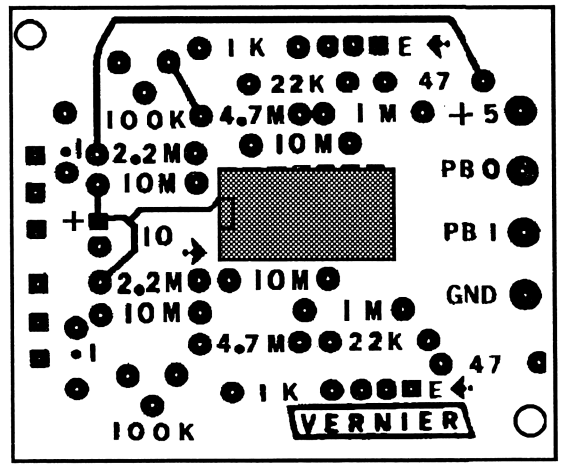
Overview: These instructions assume you are using a Vernier Software Speed of Sound Parts Kit, which includes a custom circuit board. If you are building the project using a general-purpose circuit board or a prototyping board, you should use the schematic diagram in the "Background Information" as a guide.



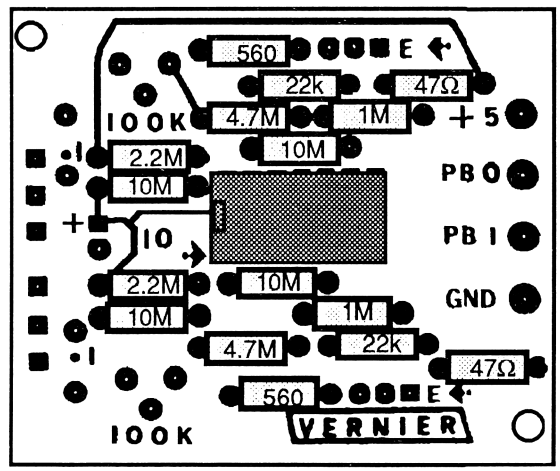
The Component Side of the Circuit Board and the Parts to be Added

☐ 1. **Check all components.** It is important to make sure you have all the materials for this project before you begin. Double-check the parts list given at the beginning of this project, and check each resistor and capacitor to make sure it is the proper value.

☐ 2. **Place the 14-pin socket on the circuit board.** One end of the socket will be marked with a notch to indicate pin #1. Place it with the notch toward the left side of the circuit board as shown in the diagram below. The socket and all other components should be placed on the side of the circuit board with all of the lettering on it. All soldering is done on the other side. Solder all 14 of the pins to the plated side of the circuit board. Do not place the op-amp IC into the socket at this time.



☐ 3. **Add the resistors to the circuit board.** There are 16 resistors in this circuit. The circuit board has the value for each resistor labeled on the component side, with two exceptions—the two 560-Ω resistors. They go in the two locations labeled "1-K" on the circuit board.[3] Use these labels and the diagram below to position each resistor on the circuit board. The parts list at the beginning of this project has the color codes listed.[4] Push the leads into the holes as far as possible. Bend the leads to hold the resistors in place. Solder all the leads. Cut off the excess leads as close to the circuit board as possible.

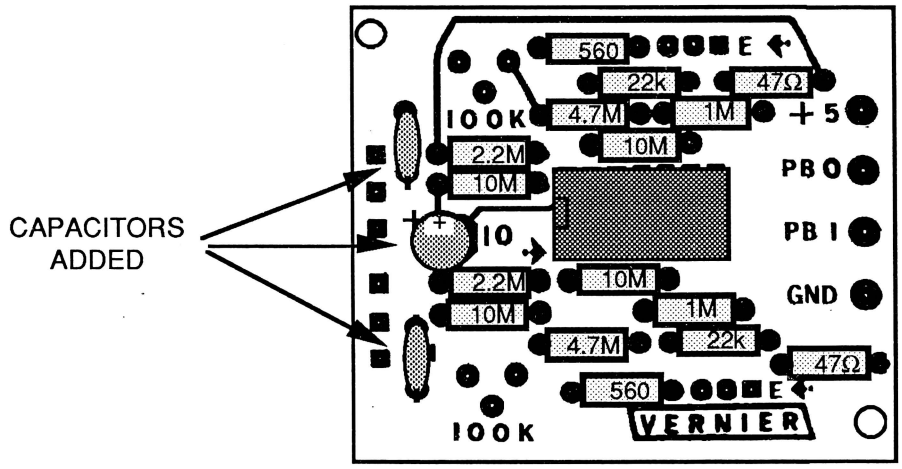


☐ 4. **Add the capacitors to the circuit.** The 10-μF capacitor has a "–" mark on it. That side goes toward the bottom of the circuit board. The 0.01-μF capacitors do not have "–" and "+" leads and can be oriented either way. Position both capacitors as shown below and solder all three components in place. Cut off the excess capacitor leads as close to the circuit board as possible.

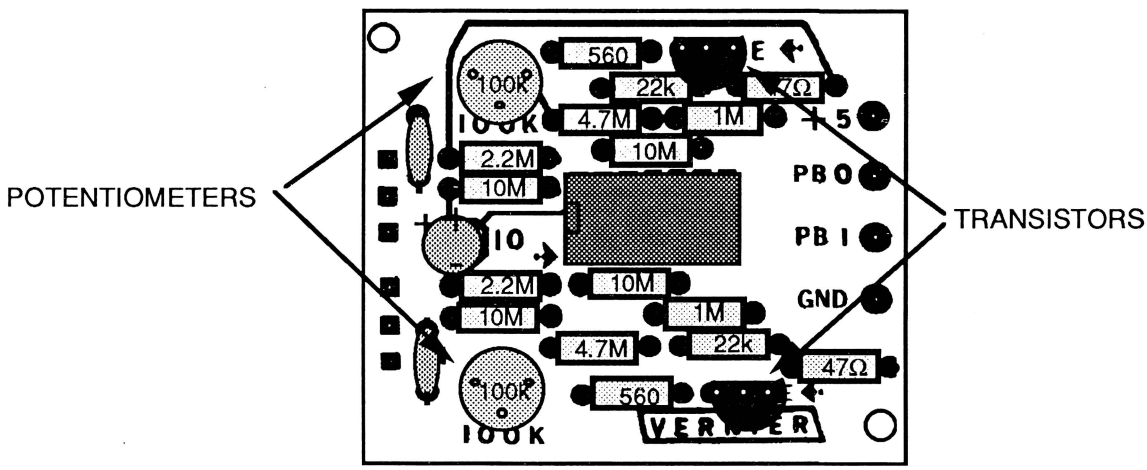
---

[3]This is the result of a last-minute circuit change to make the program work on Apple II+ computers.

[4]All of these are carbon-film resistors, except the two 1-MΩ resistors and the two 2.2-MΩ resistors. These four are precision metal-film resistors. Remember that the color code for labeling metal-film resistors is a little different. Refer to *Appendix B* if you need to learn how to read the color codes.

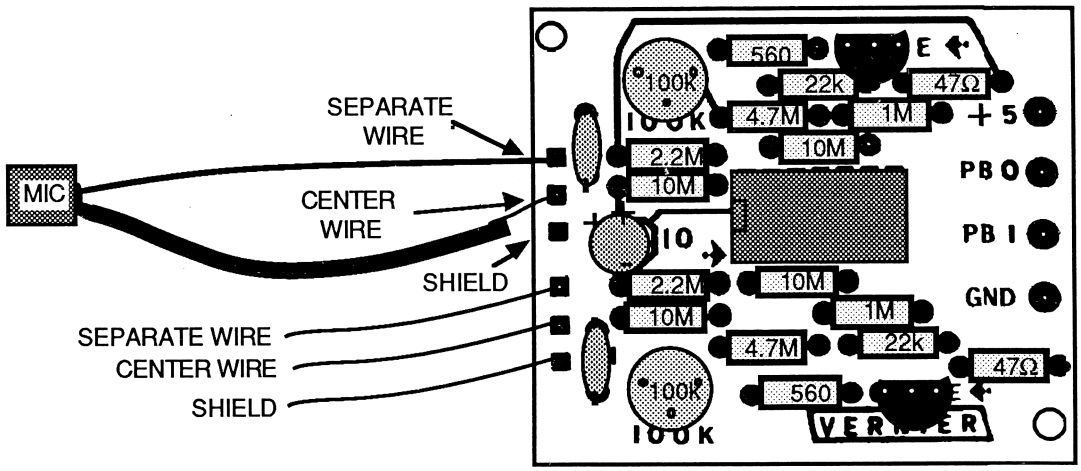


☐ 5. **Position the two potentiometers and the two transistors.** Positioning the potentiometers can be confusing. If you orient them correctly, they will slip into the three holes with almost no bending of the leads. To double-check their orientation, check for a small notch in the side of the potentiometer. It should be toward the top of the circuit board. The two transistors also must be oriented carefully. The flattened side of the transistors go toward the top of the circuit board. When soldering the transistors, it is a good idea to use a heat sink between the body of the transistor and the soldering iron. The easiest way to do this is to use an alligator clip on the transistor lead being soldered. Solder both potentiometers and transistors in place. Cut off the excess transistor leads near the circuit board.



☐ 6. **Add extra lead wire to one of the microphones.** There are three lead wires for each microphone. The separate one is for the +5V power lead. The other two leads are together; the inside wire is the signal and the outer shield is the GND lead. One of the microphones needs to have these three leads extended so that it can be placed some distance from the circuit board. Prepare the 3-conductor cable by stripping 1/2 cm of insulation off all three leads at both ends. Now solder each of the leads of the microphone to the cable. Insulate the connections so that they cannot touch. Note the color of each lead so you will know how to solder the cable to the circuit board in the next step.

☐ 7. **Connect the microphone leads to the circuit board.** The microphone with the short leads connects to the three holes near the top of the left side of the circuit board. The microphone with the extended cable connects to the three holes near the bottom left. Use the diagram below as a guide for connecting the leads.



☐ 8. **Solder the computer cable to the board.** Refer to *Appendix A* to determine which leads to use on the 9-pin or 16-pin cable to the game port. You will need to solder four leads to the circuit board: +5V, PB0, PB1, and GND. These leads connect to the holes along the right edge of the circuit board.

☐ 9. **Plug in the IC.** Carefully place the integrated circuit into the socket. Be sure to orient pin #1 at the notch of the socket (toward the left). There is a small indentation or dot on the top of the IC marking pin #1. Make sure all fourteen pins go into the socket. Be careful not to bend the pins. They can easily break.

☐ 10. **Test the completed circuit.** Turn off the computer. Plug the Speed of Sound circuit into the computer. Position the microphone with the extended cable several meters away from the first microphone. Turn on the computer and run SOUND on the *Chaos in the Laboratory Programs #1* disk. If the computer does not start up normally, turn it off and refer to the "Troubleshooting" sections below.

This project depends on the computer to keep track of time intervals. It assumes that the Apple II is running at the standard Apple II clock speed. If you have an Apple IIGS, it will be automatically slowed down at time-critical parts of the SOUND program. Therefore, the SOUND program works fine on the IIGS, even if you are using a TransWarp GS card. If you are running on an Apple IIc+, hold down the <ESC> key while booting to make it run at NORMAL speed. Similarly, if you are using any computer (other than a IIGS) with an accelerator card or chip, make sure your computer is running at normal Apple II speed (1 MHz).

When the program starts, you will be asked whether you want the times to be displayed in large digits or not. Enter "Y" or "N" and press <RETURN>. Next, choose the S - STATUS CHECK mode. With no sound, the inputs should be low. When you make a continuous noise near a microphone, its status should flutter between high and low. If it continues fluttering after the sound has stopped, the circuit is too sensitive. Adjust the 100-kΩ potentiometer so that the circuit is less sensitive. Turning the potentiometers counterclockwise makes the circuit less sensitive. If you are still having problems, refer to the "Troubleshooting" section below. If the circuit seems to be operating normally, press <RETURN> and then select R - READ TIMES from the menu.

Now try making a sound using two metal objects, two blocks of wood, or two bottles close to the microphone with the short leads. The sound source and both microphones should all be along a straight line. The computer should display a result each time you make a noise. A short, sharp sound is best. We have had our best results hitting together two pairs of pliers or tapping two bottles together. Experiment and see what kind of sound seems to work the best. You may find that making

the sound louder or softer improves the results. Even how you hold the objects you are hitting together may make a difference in performance.

Measure the distance between your two microphones and calculate the speed of sound using v = d/t. How does your result compare with the accepted value? If possible, measure the temperature of the air, so that you can compare with the accepted value for the speed of sound at that temperature. What can you do to improve the accuracy of this experiment?

Troubleshooting

If your circuit does not behave as you expect it to, the following tests are recommended to help you spot problems. In your testing and troubleshooting, remember that this is essentially a double circuit. All of the components on the top half of the circuit board and the connections on pins #8 – #13 of the IC are for microphone #1. The components on the bottom half of the board and pins #1 – #6 of the IC are used by microphone #2. This should help you zero-in on problems if one microphone is working and the other is not.

Specific Problems:

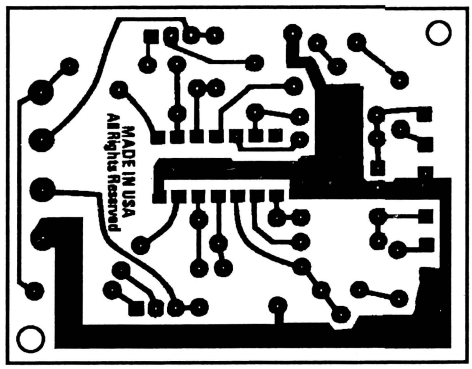
◊ If your circuit works correctly except one of the inputs remains on HIGH instead of LOW, you need to change the resistance of the resistor labeled "10 M". You need to slightly increase the resistance here. Try adding a 100-kΩ resistor in series with the resistor labeled "10 M". If necessary, add more resistance in series.

◊ If your computer displays strange graphics on the screen when you start it up with the Speed of Sound Experiment plugged in, do not panic. You are seeing the Apple IIe or IIGS "self test" in action. The Apple IIe and IIGS both have a self-test mode built into them. Any time the computer is turned on (or rebooted) with the <Closed Apple> key pressed down, the computer goes through a sequence of self-diagnostic tests. These tests take about a minute, and strange graphics are displayed on the screen as they take place. If the second microphone of your Speed of Sound Experiment is in the HIGH status when you turn on the computer, the computer will start the self test. To temporarily avoid this problem, disconnect the Speed of Sound Experiment, start up the computer, and then reconnect the Speed of Sound circuit. To permanently solve the problem, you need to change the circuit so that the second microphone stays LOW when no sound is detected. This procedure is explained in the paragraph above.

General Problems:

◊ Is your IC inserted properly? Pin #1 is marked with a dot or indentation. Is the #1 pin toward the left side of the circuit board?
◊ Have you bent one or more pins on the IC by not putting it into the socket properly?
◊ Does your op-amp get hot when you turn things on? If it is hot, you must have a serious wiring error, such as mixing up the +5V and GND lead wires from the computer. If this happens, you may need to replace the chip after you fix the wiring error.
◊ Do you have the microphones wired correctly? Double-check the diagram to make sure you have the three leads positioned correctly.
◊ Make sure you have all the resistors in the right places. Check the color codes listed in the parts list or in *Appendix B*. The resistor values are labeled on the circuit board.
◊ Is the cable to the computer getting warm? This could be a sign that the leads on the microphones are shorted somewhere. This could happen if the leads are not insulated and are touching one another. It could also happen if the connections on the case of the microphone are touching. Check this completely with an ohmmeter, if necessary.
◊ Visually double-check all of the solder joints on the circuit board. There are a lot of connections and it is easy to miss one in your soldering. Also, check for solder bridges between points on the circuit board that should not be connected. The figure below shows the back of the circuit board as it was

before you started soldering. Compare it to your circuit board to check for solder bridges. If you are still having problems, reheat all the solder joints.



The Back of the Circuit Board

If none of this helps, you can use a voltmeter to check the circuit. Connect the negative lead of a DC voltmeter to the GND connection of the circuit board. With the computer on, and the circuit connected to the game port, check the voltage at each of the following pins on the LM3900 IC.

Pin #	Voltage Should Be	Comments
2,12	about 0.5V	Is a 10-MΩ resistor connected to each of these pins?
3,11	about 0.4 V	Are the potentiometers installed correctly? Are the 4.7-MΩ resistors in the right places?
4,10	about 0.5V	Are the 4.7-MΩ resistors are in the right places?
6,8	about 4 V	Check to see if the 1-MΩ resistors are connected.
1,13	about 2.5 V	Check to see if the 2.2-MΩ resistor is connected.
5,9	about 0.2 V	Check to see if the 10-MΩ resistors are connected.
7	0 V	Check the connections on the game port cable.
14	5V	Check the +5V connection on the game port cable.

If you have an oscilloscope (or a computer software/hardware combination that acts like an oscilloscope), you can troubleshoot your circuit in the following manner: Connect the ground input of the oscilloscope to the GND of the circuit. If both inputs are malfunctioning, test each input separately. First, whistle or create some other sound at the microphone. With the oscilloscope, you should be able to see a very small signal varying around 0.5 volts at the output of the microphone. If there is no signal, then there is a problem with the microphone. There should be an amplified signal at the output of the first op-amp (pin #4 or #10 of the LM3900). If not, then there is a problem with some connection in this stage of the circuit. Double-check the resistor connections (see the schematic given in the "Background Information"). If everything here is ok, check the signal at the outputs of the second op-amp (pins #5 or pin #9 of the LM3900). There should be a square wave between 0 and +5 volts. If you do not see a square wave, then there must be a problem with some connection on the Schmitt trigger part of the circuit. If the signal is ok here, but the circuit still does not work, the problem must be in the transistor connections or in the leads to the computer.

Project Extensions

Circuit Modifications

◊ To make the Speed of Sound project more durable, you may want to mount it in a small box, with the microphones and game port cable extending out. An easier, but still useful addition is to glue the lead wires to the circuit board so that they do not break off where they are soldered.

◊ The core project has the second microphone 2 or 3 meters from the circuit board. You may be able to improve the experiment by increasing this distance. You can add additional 3-conductor cable to the microphone. To enhance the second microphone's sound-collecting ability, try using a reflector behind it to focus the sound. A parabolic shape would be best, such as a reflector from an old car headlight.

◊ If you are trying to do experiments outdoors, it may be helpful to solder some extra wire on the first microphone. You can then keep the computer and circuit inside and close to a wall outlet, and do the experiment outside.

Program Improvements

The SOUND program on the *Chaos in the Laboratory Programs #1* disk is a fairly simple BASIC program that can be improved by adding new features. The "Program Design Notes" section below explains how this program operates. It should help you in making modifications. Some suggested changes are given below. All of these modifications can be done by modifying only the BASIC portion of the code (the main program named SOUND).

◊ Add statistics to the program so that the mean time, standard deviation, etc. are calculated after a specified number of timings.
◊ Allow the timing data to be saved to memory.
◊ Allow timing data to be saved on a disk and reloaded later.
◊ Have the computer ask for the distance over which the timing is done, and then have the computer compute and display the speed of sound after each timing.

Additional Experiments

DETERMINING THE RELATION BETWEEN THE SPEED OF SOUND AND TEMPERATURE

There are two ways to do this experiment: (1) Set up the Speed of Sound apparatus in an enclosure where you can control and accurately measure the temperature; or (2) Place the Speed of Sound apparatus outside, and measure the speed of sound several times during the day as the temperature changes. Either way, make a plot of the speed of sound in air vs. temperature. See if it matches your theoretical predictions that you find in college physics texts.

USING THE SPEED OF SOUND APPARATUS AS A RANGE FINDER

Once you have found the speed of sound in air for a particular temperature, you can use the experimental apparatus as a range finder. Modify the SOUND program to compute the distance between the two microphones from the time it measures and the speed of sound that you enter at the start of the program.

MEASURING THE SPEED OF SOUND IN OTHER GASES

The experimental apparatus you have built can also be used to measure the speed of sound in gases other than air; for example, nitrogen, CO_2, and oxygen. The "Supplemental Information" section lists

the speed of sound in these and other gases. One easy way to do an experiment with other gases is to use a mylar balloon filled with Helium—the kind frequently sold for decorations.

MEASURING THE SPEED OF SOUND IN SOLIDS

You can also measure the speed of sound in solid materials. For example, try mounting the microphones on an aluminium rod. Hit one end of the rod with a mallet. Remember that because of the higher density, the speed of sound is much higher in a solid. A table of some typical values is also given in the "Supplemental Information" section.

Supplemental Information

Tips for Getting Good Results

◊ One key to the success of this experiment is to produce a consistently sharp sound directly in front of the first microphone. The sound must be loud enough to be heard every time by both microphones. Using two heavy metal objects, such as pliers, seems to work well. Some of our test sites have reported good results with bottles, blocks of wood, and billiard balls. Try different objects. Try changing the way that you hit them together. You may also want to experiment with different potentiometer settings. With experimentation, you will discover how to produce a sound that gives you consistent, reproducible times.

◊ Since many trials can be done in a short time, always repeat the timing several times. Note the pattern of values reported. Usually one time will be reported over and over, with a few strange readings which you can disregard.

◊ Since this project depends on sound for controlling the timing, too much noise in the room can cause incorrect timing. Try to keep the room quiet except for the sound used in the experiment.

◊ Sometimes an unrealistically long time is reported. This is most likely due to a weak sound level at the second microphone. The timing does not stop when it should. The second microphone may pick up another sound later (perhaps from another trial hit) and report an incorrect time. If this happens frequently, increase the sensitivity of the second microphone by adjusting its appropriate potentiometer, improve your sound source, or move the second microphone closer.

◊ A distance of 5 meters seemed to be the optimum distance for speed-of-sound measurements in our experiments. We have gotten good results from a distance of up to 8 meters, but a lot of bad timings occur at that distance (from other noises and the reduced sound level at the second microphone). The nice thing about this experiment is that this bad data is very easy to spot (it is significantly different from the usual numbers).

More on the Speed of Sound

Sound waves are mechanical waves moving through a medium by an interaction between molecules. The speed of any mechanical wave depends on the medium through which it travels. In most cases, the velocity of a mechanical wave in a gas depends on the density of the medium and its elasticity. For sound waves, the equation for the speed of sound is usually written as:

$$\text{velocity} = (B / \text{density})^{1/2}$$

where B is the *adiabatic bulk modulus* of the medium. B is roughly a measure of the compressibility of the medium. If the substance can be compressed easily, its B value is relatively low. A substance such as water, which is very difficult to compress, has a very high B value.[5]

---

[5]For a complete discussion of this equation and the physics involved in it, refer to a college physics text such as *Fundamentals of Physics*, 3rd Edition, David Halliday and Robert Resnick, John Wiley & Sons, New York, 1988.

For sound waves in air, the speed turns out to be about 343 m/s in air at room temperature (20°C). As the temperature increases, the density of the air decreases and the speed of sound increases. The increase in the speed of sound for a one degree Celsius change in temperature is about 0.59 m/s.

The speed of sound varies greatly in other media. Typical values for the speed of sound in different materials at room temperature (25°C) are given below:[6]

Medium	Speed (m/s)
air	346
hydrogen	1339
alcohol	1207
water	1498
glass	4540
aluminum	5000
iron (steel)	5200

Here is a table of values for different gases at 0°C:[7]

Gas	Speed (m/s)
air (moisture free)	332
ammonia	415
carbon monoxide	338
carbon dioxide	259
helium	965
hydrogen	1284
nitrogen	334
oxygen	316

Program Design Notes

There are four programs on the *Chaos in the Laboratory Programs #1* disk for use with this project. SOUND is the BASIC program which you may want to modify. SOUND.0, the timing portion of the program, is written in machine code. This section should not need to be modified. SOUND.SRC is the assembly-language source code for SOUND.0. LP.1200 is the machine-language code that handles the large digit display.

Here is a section-by-section explanation of the SOUND program:

Lines 2 to 95: Initialization
This section loads the machine-language program, clears the screen and keyboard buffer, and prompts user for output type (large digit or normal).

Lines 100 to 198: User Menu
This section prints the menu on the screen, giving the user the choice of checking the status of the inputs, reading actual times, or quitting the program.

Lines 200 to 298: Reading Times
This section reads the time from a machine-language routine starting when the first microphone detects a sound to when the second microphone detects a sound. It then displays the timing result in counts and seconds on the screen.

---

[6] Williams, John E., Trinklein, Frederick E., Metcalfe, H. Clark, *Modern Physics*, Holt, Rinehart and Winston, New York, 1976, p. 253.

[7] Marantz, Samuel A., *Physics*, Benziger Brothers, New York, 1969, p. 355.

Lines 300 to 390: Status Check
This section simply checks the status of the PB0 and PB1 lines to see if they are high or low, and displays the result on the screen.

Lines 600 to 665: Large Digit Display Subroutine
This section handles the printing of the large digit display.

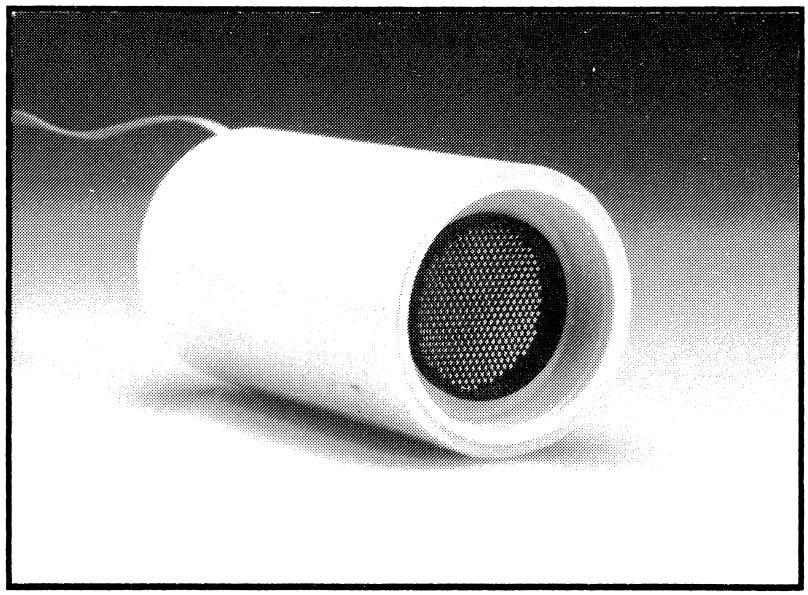
Machine Language

The source code used to produce the SOUND.0 machine-language portion of the program is included on the *Chaos in the Laboratory Programs #1* disk in the file named SOUND.SRC. Tips on using these source code files are included in *Appendix E*. The operation of the program is outlined here:

First, the computer waits until PB0 goes high. Once it gets a signal, it goes to a routine that counts until PB1 goes high. There is a slight delay here from when it gets the signal of a change in the status of PB0 until it starts timing. This delay averages 37.1 µs and this correction factor is built into the BASIC part of the program (variable CF). Once PB1 goes high, the counting is stopped and the number of counts is sent to the BASIC portion of the program. Each count represents 16 cycles of the computer's clock. There are 1,023,000 clock cycles in a second; therefore, each count is about 16 µs. This number is accurate for all Apple II computers running at normal speed.

The following variables are used in the SOUND program:

AB	Memory location used in large digit display
AC	Character being printed in large digit display
CA	Memory location used in large digit display
CF	Correction factor = 37.1 µs
CH	Loop control variable
CP	Counts per second = 63,937.5
CT	Number of counts
D$	Applesoft prompt when loading machine code
GS	Temporary variable used in setting Apple IIGS speed
H	Horizontal starting point for large digit display
HZ	Memory location used in large digit display
I	Loop control variable
KB	Holds the last keyboard entry
KC	Variable telling computer that the last keyboard entry was checked
L$	User input string variable
LW	Constant used in large digit display
PR$	Blank line used for an invalid time
Q$	User input string variable
TI	Actual time reported
V	Vertical starting point for large digit display
VZ	Memory location used in large digit display section



Project 13

Ultrasonic Range Finder

by Peter Bruecken

This project involves connecting your Apple II computer to an ultrasonic range finder. This is the device used on many automatic-focus cameras. The Range Finder is connected to the 16-pin internal game port of an Apple II+, IIe, or IIGS. The Range Finder sends out a pulse of ultrasound and the computer measures the time it takes the ultrasound to travel to an object and back. The computer can then calculate distances almost instantly, even as an object moves! The power and speed of the computer can be used to display the distance and graph the data.

Project Requirements

Computer

This project uses the annunciator output lines on the 16-pin Apple II game port. These lines are available on the Apple II+, IIe, and IIGS. This project cannot be done on other models of Apple II computers.

Background Knowledge

- Electronics: This project requires careful soldering of wires and components on a circuit board. Previous experience with soldering is recommended.

- Programming: No programming is required for the core project. A few of the project extensions require a knowledge of BASIC.

Parts

- ☐ Polaroid 615077 transducer & Texas Instruments SN28827 circuit board
- ☐ electrolytic capacitor (2200 MFD, rated at 6 volts or more)
- ☐ 4-conductor ribbon cable
- ☐ 16-pin DIP header and cap
- ☐ 3.9-kΩ, 1/4 watt carbon resistor
- ☐ PVC pipe, with end caps and mounted transducer diaphragm
- ☐ transducer clips (attached to the transducer)
- ☐ (2) 7-cm long wires

All of these parts are available as a kit from CAPE-tech, Inc., P.O. Box 115, Wynnewood, PA 19096-0115 phone (215)-789-0924 for $75 as of January 1991 (Order Code M1-K). CAPE-tech is the proprietary owner of the design presented in this chapter. CAPE-tech has a special student price of $68 for their parts kit for students only if prepaid (Order Code M1-SK).

Tools and Supplies

Soldering iron (low power), solder, electrical tape, wire cutters, wire strippers, a small amount of silicone sealant. A multimeter may be helpful.

Background Information

Measuring distance is a simple process. Usually, we need to compare the distance in question with some standard, such as a meter stick or tape measure. The Apple II can measure distance using the speed of sound as the standard. The advantage of using the computer for distance measurement is that the distance can be display or graphed almost instantaneously, even while the object is moving.

Sound moves through air in the form of vibrations called waves. Waves move through air without taking the air with them. This allows sound to move quite rapidly using very little energy. Sound waves vibrate the air over and over again a certain number of times per second. This is the wave's *frequency* and it is measured in waves per second. The number of waves per second is expressed in units called *hertz* (Hz). Our ears are sensitive to waves if the waves have frequencies between about 20 Hz and 20,000 Hz. Vibrations with a frequency higher than 20,000 Hz are called *ultrasonic*. The Range Finder uses waves to measure distance. Two convenient aspects of these "sound like" waves are that they cannot be heard by the human ear[1] and they travel at a nearly constant speed. Ultrasonic waves in air travel about 343 meters per second at room temperature (20°C). Using this figure, and the computer to control the Range Finder and measure time, we can measure distances. We have the computer start the ultrasonic sound wave from the Range Finder and time how long it takes the wave to travel to an object and return again. The computer keeps track of the time. We can calculate the distance to the object and back by multiplying the time for the complete trip by the speed of the wave and then dividing by two:

distance = (time * speed of ultrasound) / 2

The division by two is because the wave must make a round trip from the Range Finder to the object and back. This is exactly how the Range Finder works.

[1]This is nice because you will not be bothered by sound from the Range Finder as you do experiments. An interesting topic for investigation might be: Do other animals hear the ultrasonic sound from the Range Finder? Ultrasonic dog whistles and ultrasonic rodent and insect repellants have been on the market for years.

The Range Finder has many uses. Alarm systems use Range Finders as motion detectors. Some Range Finders are put in cases with digital readouts and used as "tape-less" tape measurers. These can be found in hardware stores for carpenters and builders.

The Range Finder has two parts: a gold foil diaphragm called a transducer[2] and a circuit board. The diaphragm is made of very thin, gold foil and is encased in a black, metal grid. The black grid protects the diaphragm and provides a means to connect it with the circuit board. It also insulates the foil, which carries 150 to 300 volts. The transducer is made to vibrate at 50,000 times per second (50 kHz) to create an ultrasonic signal. It also picks up the return of the ultrasonic signal and converts it back into electric energy. In this way, it acts like a speaker and a microphone. The transducer may look familiar to you because it is the auto-focus transducer from a Polaroid SX-70 camera.

The circuit board includes all the circuitry to operate the transducer. Figure 1 is a diagram of the five functional parts of the circuit board and transducer. The board must produce a high-frequency, high-voltage signal on command from the computer. It also must produce a steady high voltage for the transducer. Finally, the board must detect when ultrasonic echoes are being received by the transducer and report this in a way that the computer can interpret. This board is made to operate the transducer in the type of electrical environment that can be supplied by the Apple II computer.

The transducer requires 150 volts of electric potential to be in the "ready" state. The circuit board produces the 150 volts from the computer's 5-volt DC power supply by using a transformer. Since transformers will not work with direct current, the circuit board changes the +5-volt computer supply to an oscillating voltage using a 50-kHz oscillator. This oscillating signal is connected to the transformer, which converts it to magnetic energy and reconverts the magnetic energy to high-voltage electric energy. This alternating electrical energy is then changed back to constant, non-oscillating electricity of higher voltage by a rectifier. A *rectifier* is four diodes connected to direct all positive current in one direction and all negative current in the other direction. After the current goes through the rectifier, it is applied to the transducer to put it in the "ready" state. The trade of low voltage (5 volts)/high current to high voltage (150 volts)/low current conserves energy. This also is why the high voltage used in this project is not dangerous—it can never supply very much current, or do much damage.

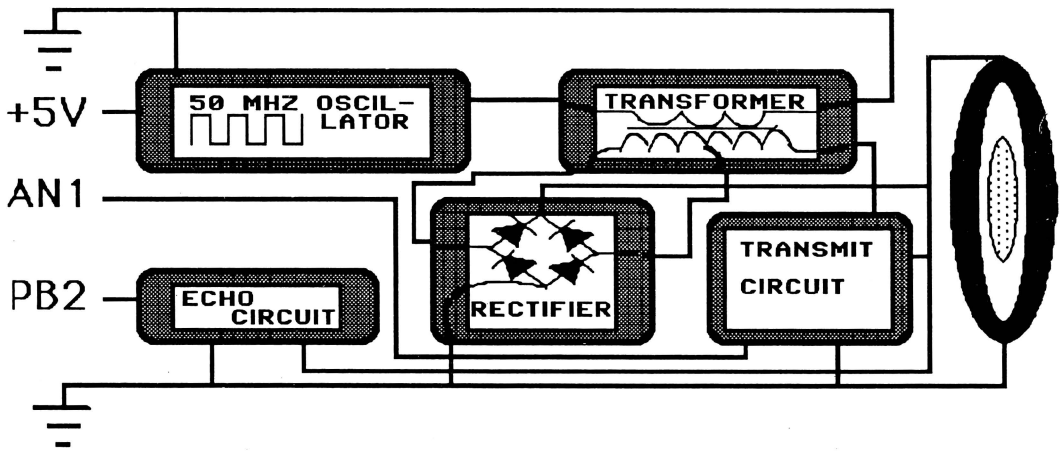


Figure 1 - The Functional Parts of the Range Finder

Figure 2 is a voltage vs. time graph showing the functions of the board. When the board receives the signal from the computer's AN1 line to activate the transmit circuit, it sends a 150-volt signal (oscillating at 50 kHz) to the transducer to vibrate the diaphragm, launching the ultrasonic signal into the air. The board's echo circuit then waits for the reflected signal to vibrate the diaphragm. When a

[2]A *transducer* is any device that converts one form of energy to another.

vibration is detected, the echo circuit indicates this by making its output go high briefly. This output is connected to the PB2 input line of the computer.

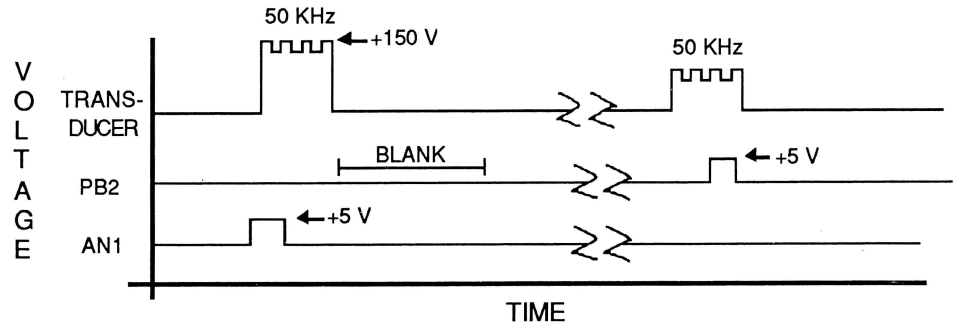


Figure 2 - The Signals Used by the Range Finder

The echo circuit ignores vibrations of the diaphragm for a period of time after transmission. This period is necessary to let the diaphragm settle down from its "speaker" function and change to its "microphone" function. The residual vibrations from transmission can falsely trigger the echo circuit. The period of about 2.38 milliseconds (0.00238 seconds) after transmission is ignored by the echo circuit to let the transducer stop oscillating. This time is marked "BLANK" in figure 2.

There are two additions to be made to the Range Finder circuit board to make it compatible with the Apple II game port: a filter capacitor and a pull-up resistor. The filter capacitor is connected between the +5V and GND lines from the game port to "smooth out" the power supply voltage. This circuit board needs an extremely brief rush of electrons to start the transducer vibrating and this tends to momentarily drop the voltage on the 5-volt line in the computer. This capacitor eliminates this problem. The pull-up resistor is used on the PB2 digital input line from the Apple II game port. It is used to keep the voltage on the game port line high until the circuit board echoes its signal back to the computer. Without this resistor, the PB2 line on some Apple II computers tends to "float" high and low.[3]

The useful range of the device is 0.40 cm (1.33 feet) to about 10.7 meters (35 feet). If an object is too close, the Range Finder will not measure the distance, because the returning ultrasound will come back during the blank echo period. For this circuit, the minimum range is about 0.4 meters.[4] Typically 10 or 11 meters is the maximum distance at which a return wave can be detected. If the object is too far away, the return signal will be too weak to be detected. The sampling rate can also limit the maximum range. If the program tries to take measurements too often, and the object is far away, the program may try to start a new measurement before the previous wave has returned.[5]

The Range Finder can pick up the position of the wrong object if the area contains other objects in the path of the signal, so be careful to operate it in a place where it will pick up only what you wish. Recognizing these characteristics is essential for success with the Range Finder.

Two programs on the *Chaos in the Laboratory Programs #1* disk can be used to operate the Range Finder. MOTION is a fairly complete program that will display the distance in large digits, make graphs of distance vs. time, or even velocity vs. time. The program DISTANCE is a simple starter

---

[3]The reason we use the PB2 line instead of the PB0 or PB1 inputs involves pull-down resistors. The other two input lines (PB0 and PB1) have resistances connected on some of the newer Apple II computers. These internal resistors make PB0 and PB1 incompatible with the Range Finder.

[4]The blank echo period is 2.38 milliseconds. Ultrasound can travel about 343 m/s * 0.00238 s = 0.816 m in this time. Since the ultrasound has to travel to the object and back, the minimum range is about 0.4 meters.

[5]One of the programs supplied on the *Chaos in the Laboratory Programs #1* disk takes readings 40 times a second. In 1/40th of a second, ultrasound can travel only about 8 meters. As a result, the maximum range at this sampling rate is about 4 meters. Remember that the wave has to make a round trip from the Range Finder.

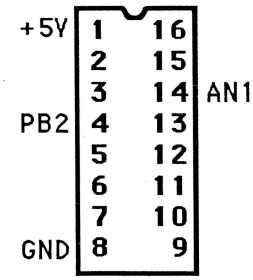
program that you may want to use if you want to write your own programs to operate the Ultrasonic Range Finder.

Step-by-Step Instructions

Assembly of the Range Finder involves mounting some components on the circuit board, soldering the cable to the circuit board, connecting the board to the transducer, and mounting the circuit board and transducer in the PVC pipe. These instructions assume you are using the parts kit from CAPE-tech.

SOLDERING THE CABLE TO THE CIRCUIT BOARD:

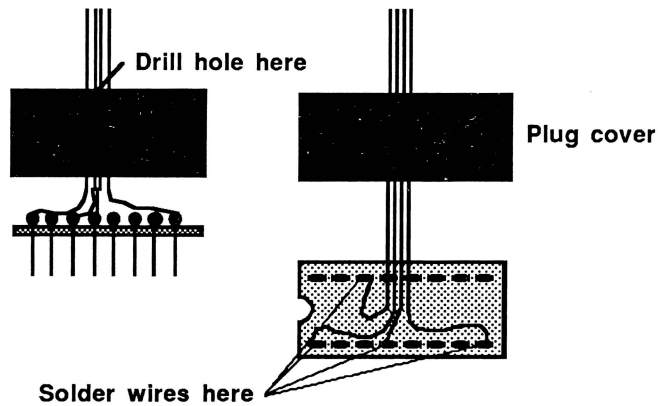
☐ 1. **Prepare the cable for the 16-pin internal game port.** You will use four pins from the 16-pin internal game port: +5V (pin #1), GND (pin #8), AN1 (pin #14), and PB2 (pin #4). Since the annunciator output line is needed, this project cannot be done using a 9-pin game port. Refer to the figures below to identify the connections.



The Apple II 16-Pin Game Port

The plug comes in two pieces: a DIP header (plug) and the cover. You must drill or melt a hole in the center of the top of the cover for the cable. Feed the cable through and separate about 1 cm of the four wires. Strip about 2 mm of insulation from the ends of each wire and melt a small amount of solder to each wire end. Melt a very small amount of solder to the tops of pins 1,4, 8, and 14 on the plug. *Do not let the plastic part of the DIP header get too hot.* If the pins get too hot, they will melt the plastic part of the DIP header.[6] Gently heat the wires and pins until the solder melts and the wires stay on the pins. Be careful not to cause a solder bridge between two adjacent pins.

After the solder cools, label the four wires "+5V", "GND", "AN1", and "PB2".[7] Now, fill the inside of the cover with the silicone sealant and press it on the DIP header. This will keep any tension on the cable from breaking the solder connections.



☐ 2. **Prepare the resistor and capacitor.** First, identify the leads of the capacitor. These capacitors have a "–" marking the negative lead, often with an arrow pointing to the negative lead. The

[6]We find it helpful to put a heat sink on the pin being soldered. Even though the heat sink is on the opposite side of the plastic from the solder, it helps keep the metal pin cooler.

[7]If you do not label the wires now, you will have to use an ohmmeter to identify which wire goes to which pin after you glue the cover on.

other lead is the positive lead. Bend the positive lead of the capacitor so it comes back parallel to its axis to about the middle and bend the remaining length down as shown in figure 3. Bend the negative lead down parallel to the positive lead. This should make the parallel parts of the two leads about 1.8 cm apart. Now bend one lead of the resistor at a right angle as shown. Twist this end of the resistor and the +5-volt line from the computer onto the positive end of the capacitor and solder them. Solder the ground wire from the computer to the other end of the capacitor. Solder the PB2 wire from the computer to the other end of the resistor (figure 3). Cut the leads so they are all flush as shown in the diagram.

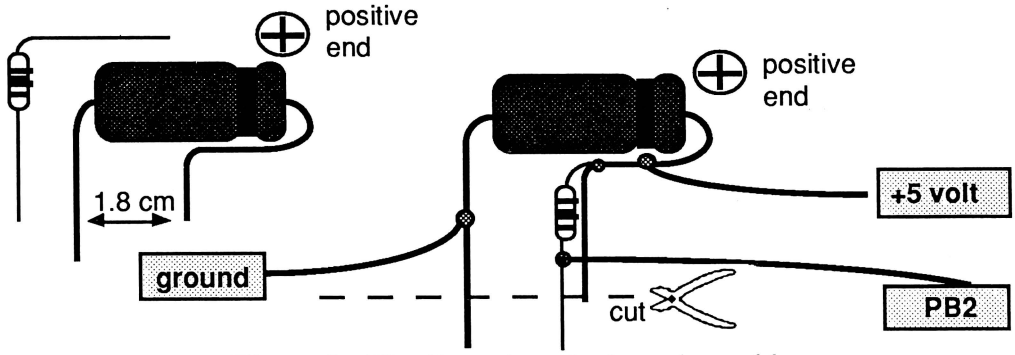


Figure 3 - The Capacitor-Resistor Assembly

☐ 3. **Identify the connection points on the circuit board.** Turn the circuit board over to the side with the components. Identify the 8 hole next to "J1" on the circuit board as shown in figure 4. You will be using holes #1, #2, #5 and #8. The four holes that will be used are shown in the figure.

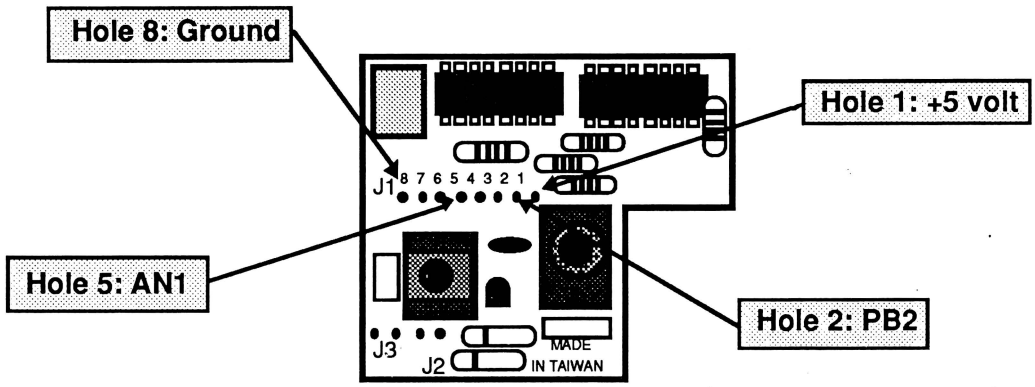


Figure 4 - The Connection Points on the Circuit Board

☐ 4. **Solder the capacitor-resistor assembly to the circuit board.** Your capacitor-resistor assembly can now be soldered to the circuit board. As you lower it onto the board, the leads of the capacitor and resistor can be gently bent to match the proper numbered hole in the board. The holes in the circuit board are already solder filled. You should be able to gently heat the solder that is already in the hole, insert the lead and then withdraw the iron and let the solder cool. The positive end of the capacitor should be carefully soldered into hole #1. The other end of the capacitor and the free end of the resistor may have to be slightly bent to get the positive end to fit into the hole. The free resistor end can now be bent and soldered into hole #2. The negative end of the capacitor should then be bent and soldered into hole #8.

When soldering, if the previously connections to the game port wires come unsoldered, a "heat sink" can be clamped onto the leads between the circuit board end and the soldered end. A needle nose pliers or any other metallic clamp can be used for this.

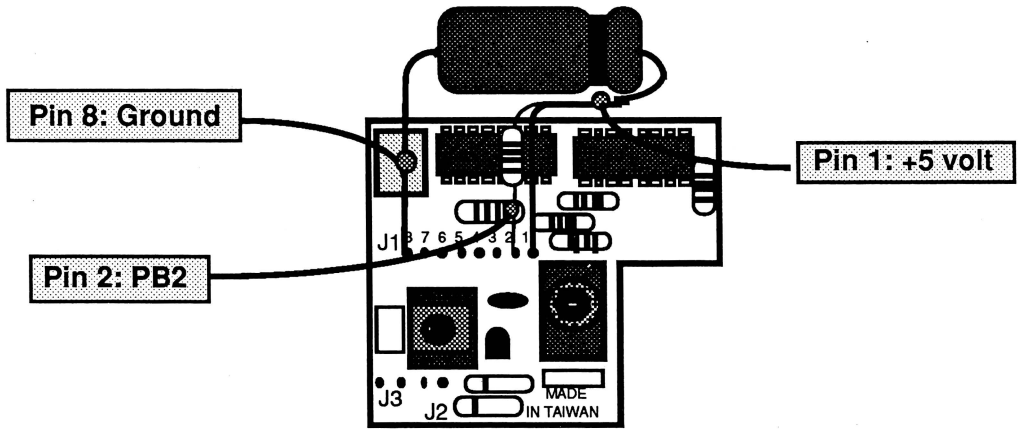


Figure 5- The Capacitor-Resistor Assembly Attached to the Circuit Board

☐ 5. **Solder the AN1 line to the circuit board.** Find hole #5 on the circuit board. Solder the remaining AN1 line from the game port cable to this hole as shown in figure 6. If the wire has loose strands, twist them together and melt a small amount of solder on the end before soldering the wire into place. If the capacitor-resistor assembly is in the way, the soldering iron can be placed under the board to melt the solder from the other end. Check to make sure the correct hole (#5) is used.

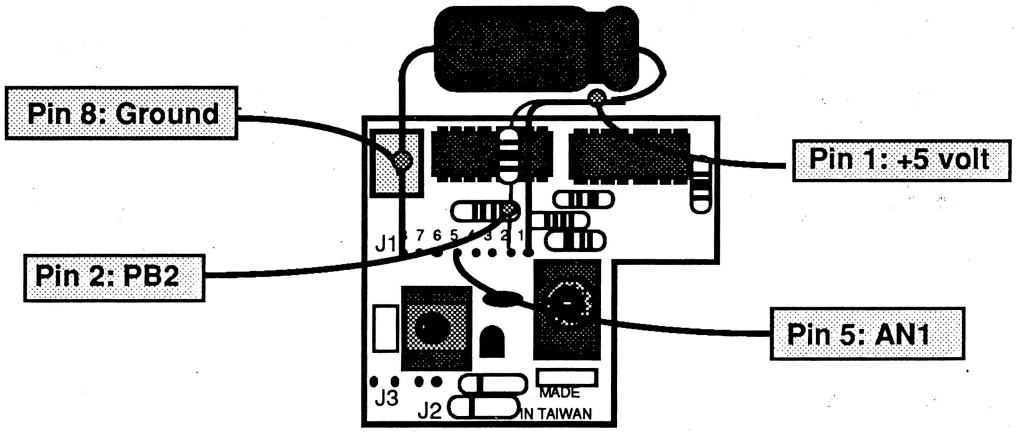


Figure 6 - The AN1 Line Connected to the Circuit Board

☐ 6. **Bend the capacitor and resistor assembly into place.** Once the four connections are made to the board, bend the capacitor-resistor assembly as close to the center of the circuit board as possible. These components have to fit into the round tube, so they have to be as low and close to the center of the board as possible without touching any of the metal parts of the other components on the board.

☐ 7. **Double-check your connections.** Make sure the four connections are correct: The PB2 line from the computer should be on the circuit board end of the resistor. The AN1 line should be alone in hole #5 of the circuit board. The +5-volt line should be on the positive end of the capacitor and in hole #1 of the circuit board. Finally, the ground line from the computer should be on the negative end of the capacitor; that is, in hole #8. Make sure there are no places where leads are touching other leads and causing short circuits.

☐ 8. **Position the capacitor.** Tape the capacitor in place so that it doesn't move if the sonar ranger is bumped. The assembly is quite rugged, but the capacitor is rather large and should be taped down.

☐ 9. **Prepare the transducer wires.** There are two clips for connecting wires to the transducer shell mounted on the transducer. Remove these clips and solder them to the two short wires. Solder the other ends of these wires to the J2 and J3 holes on the component side of the circuit board. Cut off the excess lead wires at the circuit board. Do not place the clips back on the transducer at this time.

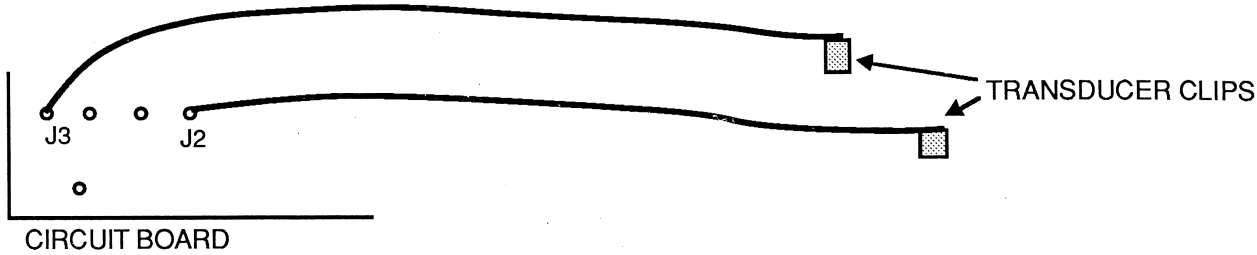


Figure 7 - Preparing the Transducer Wires

PREPARING THE PVC PIPE:

The PVC pipe assembly comes in five pieces: the end piece with the transducer diaphragm, two half-circle "wedges" for inside the pipe, the pipe itself, and a blank end piece. The wedge pieces are slipped into the pipe to hold the circuit board after it is wired. One of them has a slot cut in it to accommodate one edge of the board. The blank end piece has a slot to accommodate the ribbon cable.

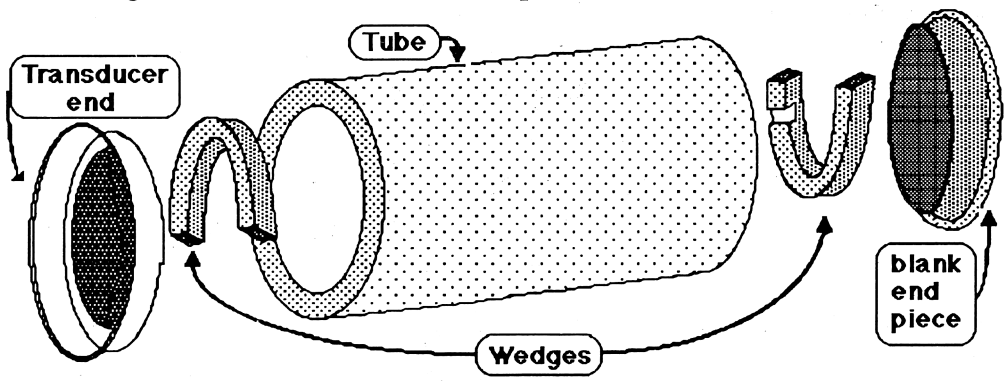


Figure 8 - The PVC Pipe

☐ 10. **Insert the first wedge.** The front wedge is the one without the slot. It should be inserted into one end of the pipe. This end is now the front end. The wedge may have to be gently compressed to fit into the pipe. After it is inserted, slide it about 2 cm from the front edge of the pipe.

MOUNTING THE CIRCUIT BOARD AND TRANSDUCER IN THE PIPE:

☐ 11. **Mount the circuit board in the pipe.** Carefully slide the circuit board into the back of the tube. The capacitor may have to be gently bent to fit into the tube. Make sure the leads of the capacitor do not touch any of the connections on the circuit board. The transducer leads with their clips should extend out of the front of the tube.

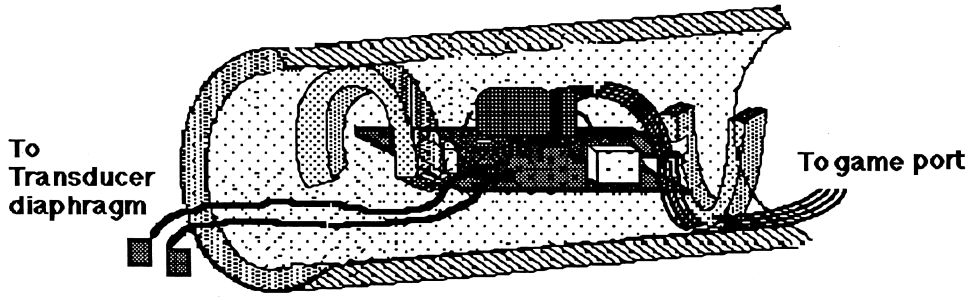


Figure 9 - Mounting the Circuit Board in the Pipe

☐ 12. **Add the second wedge.** Once the circuit board is placed in the tube, guide the computer lead to the bottom of the tube and out the back. Gently compress the rear wedge (with the slot) and push it into the tube with the computer lead wires between it and the tube. The wedge will act as a cable restraint, so forces on the cable will not be transmitted to the solder connections on the board. Leave some slack in the cable between the circuit board and the wedge. Carefully push the rear wedge into the board, making sure the slot engages the small section of the rear of the circuit board.

☐ 13. **Connect and mount the transducer.** Connect the clips on the wires from the circuit board to the transducer as shown in figure 10. It doesn't matter which wire goes on which clip. Slide the transducer into the end of the pipe and either glue or tape it in place. On the other end of the pipe, align the slot in the blank end piece with the cable coming out of the tube and push the blank end piece into the tube. Tape or glue it in place.

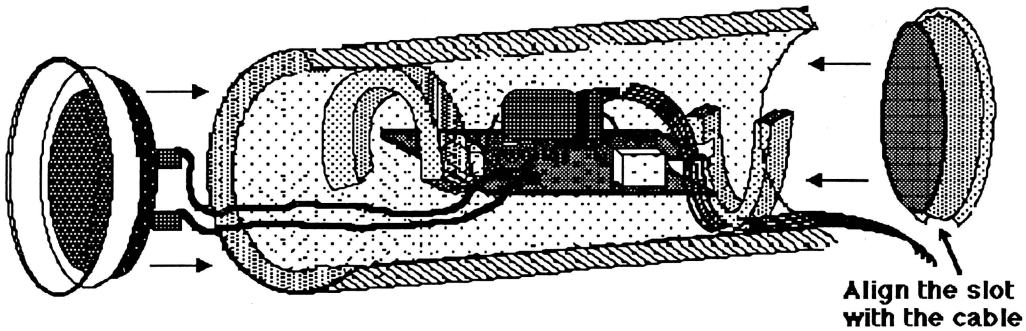


Figure 10 - Final Assembly of the Range Finder

☐ 14. **Test the Ultrasonic Range Finder.** Remove the cover of your Apple II computer and carefully plug the 16-pin plug into the game port socket inside the computer.[8] The notch on the plug should match pin #1 of the socket. Place the *Chaos in the Laboratory Programs #1* disk in the disk drive and turn on the power. Run the program MOTION. When the menu appears, choose S - SHOW DISTANCE ON THE SCREEN. The program should begin displaying the distance from the Range Finder to the nearest object. Remember that the distance must be between 0.4 meters and 8 meters for the Range Finder to work properly. Press <RETURN> when you want to examine the other options on the menu. If the system is not working properly, refer to the "Troubleshooting" section below.

> Do not touch the inside of the transducer or the circuit board while the Range Finder is operating. A high voltage is applied to the transducer. It will not harm you, but it will make you jump.

If you want to make graphs of motion, choose R - RECORD AND GRAPH DISTANCE DATA. This choice allows you to create a distance vs. time graph and at the same time store data in memory. You have four choices for the data collection:

1 - 10 SEC, 40 SAMPLES/SEC, 4M
2 - 20 SEC, 40 SAMPLES/SEC, 4M
3 - 40 SEC, 40 SAMPLES/SEC, 4M
4 - 80 SEC, 20 SAMPLES/SEC, 8M

[8] A game port extender, which brings the 16-pin game port socket outside the computer, makes this connection a lot easier.

The three values listed for each choice are the time of the data collection, the number of samples taken each second, and the maximum distance that can be measured.[9] Choose one of the four and press <RETURN>. The program then sets up a graph and begins plotting distance vs. time.

After you have collected data in this way, you can regraph it as a distance vs. time graph or use the same data to plot a velocity vs. time graph. To do this choose either D - DISTANCE GRAPH OF PREVIOUS DATA or V - VELOCITY GRAPH OF PREVIOUS DATA. These modes graph the data collected previously, but allow you to choose the starting and ending values for each axis. This allows you to zoom in on the section of a graph of interest.

Troubleshooting

◊ If your computer won't boot and continually resets with the device plugged in, turn off the computer immediately. Make sure the wires are not shorted on the plug or on the circuit board. Check the resistance between pins #1 and #8 on the plug. If it is a short (0 resistance), then the wires are crossed and this will reset the computer.

◊ If the program appears to be running but the device doesn't work, try running the program while carefully listening to the end of the transducer. You should hear a faint "click" when the program begins to take data. If you don't hear this click, the device is not sending out ultrasound due to improper wiring of the AN1 line. Check the voltage between pins #1 and #8 on the circuit board. If it is not 5 volts, there is a wiring problem from the Apple computer. Also check resistance or continuity between AN1 (pin #14 on the Apple game port plug and pin #5 on the circuit board).

◊ If the device clicks when the program begins, but no data appears while the program is running, the device is generating ultrasound and has the proper voltage. The problem is probably in the PB2 line. The transducer is producing ultrasound, but the reflected wave is not being detected or the computer is not being notified. Be sure you have the wire on the proper pin #4 of the game port and that the resistor is wired correctly and not touching any wires on the circuit board.

◊ If the device works fine but stops in the middle of a run, it may be due to acoustical problems. If an object gets closer than 1.33 feet, the return signal will not be detected and the program is waiting for the return. Gently tap the front of the device to simulate a return signal. This may happen if the object gets too far away and the return signal is lost. Again, tap the front of the transducer to allow it to continue.

◊ If you are using an Apple IIe or IIGS, make sure that nothing is plugged into the 9-pin game port.

Project Extensions

Program Improvements

Here are some improvements you might make to the BASIC program MOTION. The actual data collection in this program is done by a machine-language program that you should not try to change unless you have had experience in 6502 machine-language programming. The graphs and displays are done in BASIC and can be changed. The "Program Design Notes" section describes another program named DISTANCE, which is short and simple and is included as a starting point for new programs using the Range Finder.

---

[9]When the sampling rate is 40 samples/second, the maximum distance is limited to about four meters. This is because the ultrasonic sound must complete the trip from the Range Finder to the object and back before the program attempts to take a new reading. The speed of ultrasound is about 343 m/s, so in 1/40th of a second ultrasound can travel d = v * t = 343 m/s * 1/40 = 8.6 meters. Remember that the waves have to make the trip to the object and back in this time interval. As a result, the farthest away an object can be and still allow the waves to be back in time is about 4 meters. When the rate is slowed to 20 samples/second, the maximum distance can be about 8 meters.

◊ Change the value used for the speed of sound. The program assumes that the speed of sound is 343 m/s, which is the speed of sound in dry air at 20°C. This value is assigned in line 3001 of the program. Change this value if the conditions are such that the speed of sound is different.

◊ As written, the program displays and graphs everything in meters. If you prefer, you could modify the program to display values in feet, yards, centimeters, or some other distance units.

◊ Have the program save the position, velocity, and time data on a disk.

◊ Modify the SHOW DISTANCE part of the program so that it becomes an alarm system. If an object moves into the area covered by the Range Finder, it should start the computer beeping.

◊ The accuracy of the Range Finder can be improved by adding an initial calibration routine. This routine would allow you to adjust the program so that readings match a known distance. This calibration would correct for changes in the speed of ultrasound due to temperature changes. These calibration values could be saved to a calibration file that matches your Range Finder to your computer and room conditions at the time.

Additional Experiments

◊ Corner reflectors can be used to direct the ultrasound back to the Range Finder most effectively. A corner reflector can be made by cutting a corner of a cardboard box and using it as a reflector. The geometry of such a reflector causes any wave that strikes it to be reflected straight back to where it came from. Mount the corner reflector on the object being studied. See how much this improves the results you get. Use the S - SHOW DISTANCE ON THE SCREEN mode of the MOTION program to determine the maximum distance you can measure using a corner reflector.[10]

◊ The Range Finder can be mounted on a computer-controlled car and used as a navigational device. It can then be used to keep the car from colliding with things.

◊ The Range Finder can be used as a motion detector for an alarm system.

◊ Fast motions like free fall, radio-controlled model cars, and runners can be measured.

◊ Temperature could be measured by focusing the Range Finder on a fixed distance and measuring the time for an ultrasound wave to bounce back. The speed of ultrasound increases about 0.6 meters per second per Celsius degree. The measured distance will vary according to the temperature. Careful calibration would be necessary.

◊ Mount the Range Finder on a robot and use it as a means of navigation.

Circuit Modifications

◊ Add a longer cable to the Range Finder. For many experiments, it is nice to be able to move the Range Finder around quite a lot. The cable supplied in the kit is fairly short. Add extra wire to the game port cable. Two meters of wire should work well.

◊ The accuracy of the Range Finder can be improved by adding a temperature probe (connected to the game port) and adjusting the speed of sound value used by the program according to room temperature. The temperature sensor could be the Thermistor Temperature Probe (Project 1 in this book) or the IC Temperature Probe from *How to Build a Better Mousetrap*.

---

[10]This mode is recommended because it does not limit the range to 8 meters (or 4 meters) as the other modes of this program do.

Supplemental Information

Program Design Notes

There are two programs on the *Chaos in the Laboratory Programs #1* disk that can be used to operate the Range Finder. The program MOTION is the program that we recommend you use in testing your Range Finder. It is a fairly complete ultrasonic motion detector program that includes large digit display of distance and graphing. The other program is named DISTANCE. This is a very short BASIC program intended to be used as a starting point for programmers who want to write their own programs.

THE DISTANCE PROGRAM

The DISTANCE program simply displays the distance read by the Range Finder. The program begins by loading the machine-language subroutine DISTANCE.OBJ at address $4000. The speed of sound used to calculate the distance is stored at line 15 in the variable SOS. As shipped, the program uses the value 343 m/sec. This is the speed of sound in dry air at 20°C. The number 18 is poked into memory location 255 to set the delay between readings. This will give the ultrasonic pulse enough time to travel about 16 meters (8 meters out and 8 meters back), which is the maximum range of the Range Finder. Decreasing this value will cause the readings to be taken faster, but will limit the range. The machine-language program is then called. The machine-language program sends a signal to the Range Finder to generate an ultrasonic pulse, and then cycles through a loop while waiting for the pulse to return. The number of times through the loop is counted and used as a measure of time. When a return pulse is detected, control is returned to BASIC. The count is calculated and stored as the variable CTS. For program changes in BASIC, this is all you really need to know. The count is a measure of the time it took the ultrasound to travel from the Range Finder to the object and back. The distance (variable D) is derived by multiplying the counts by 0.5 * the speed of sound * $1.0753 * 10^{-5}$. The distance is then truncated to the nearest millimeter and displayed. This distance should be accurate to 1.8 mm. The accuracy depends on the speed of sound, which changes slightly with temperature, air pressure, and humidity. You can try changing the speed of sound (variable SOS) if you find the readings are inaccurate.

More information for those familiar with machine-language programming:

The number of cycles is stored in memory locations 8 and 9. The byte in location 9 is the high byte, so it is multiplied by 256. Each cycle through the loop represents a wait of 11 machine cycles or $11 / 1.023 * 10^6$ sec = $1.0753 * 10^{-5}$ seconds. The maximum number of loops to wait before giving up is poked into memory at location $FF (255). This number actually limits the high byte, so the maximum number of cycles is the value stored in $FF times 256. Lowering this number allows more readings/second to be taken, but also reduces the range, since it waits less time for the pulse to return. The range is equal to this number divided by 2.137. Each call to the machine language takes the same amount of time, regardless of the distance. This can be important if you need to know the time between readings. The MOTION program uses the time between readings to calculate velocity. The machine language is called DISTANCE.OBJ. The source code for it is called DISTANCE.SRC, and it is also included on the disk. Refer to *Appendix E* for information on how to use these source code files.

THE MOTION PROGRAM

This is a much more complicated program—actually a set of programs. The program MOTION initializes variables, loads assorted machine-language programs, moves memory around, and runs the program M. The only thing that can be easily changed is the speed of sound. If the distances reported by the program are wrong, you can change this variable to correct the error. Line 3001 of the MOTION program sets the speed of sound (variable SOS). The value used initially is 343 meters/second. To change it, just retype this line with the new value and save the program as MOTION. The variable OD allows you to set an *offset distance*. This is a number to add to each

distance reading. It is initially set at 0, but you may want to change this if all distance readings are off by a fixed distance.

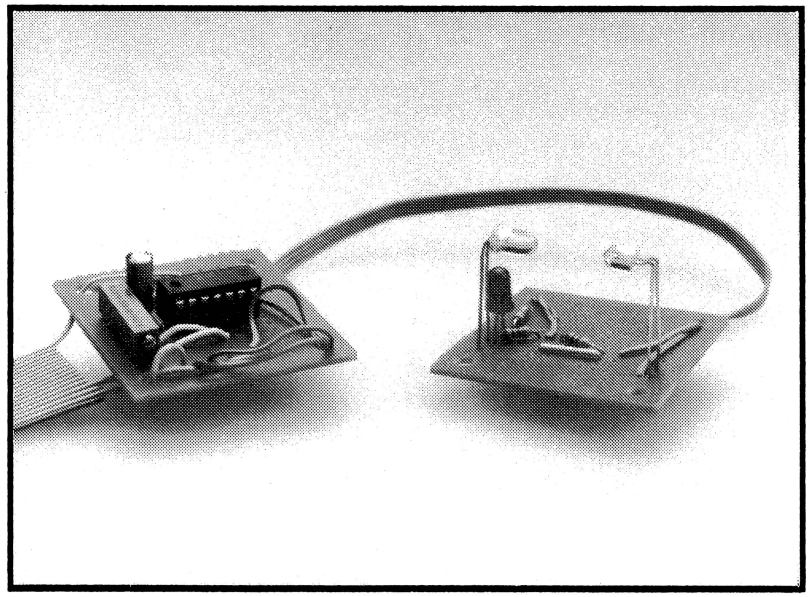
References

Ciarcia, Steve. "An Ultrasonic Ranging System." *Byte*, Oct. 1984, p. 113. This article includes a complete description of how the Ultrasonic Range Finder works, more information on the transducer, and instructions for building a stand-alone rangefinder.

Illingworth, Bruce. "The Polaroid Ultrasonic Rangefinder." *Electronic Instrumentation and Modern Optics*; Woodrow Wilson National Fellowship Foundation Institute on High School Physics, P.O. Box 642, Princeton, NJ 08542, 1985, p. 206.

Thornton, Ronald K., and David Sokoloff. "Learning Motion Concepts Using Real-Time Microcomputer-Based Laboratory Tools." *American Journal of Physics*, September 1990, p. 858.

Risley, John. "A Comparison of Range Finder Software." *The Physics Teacher*, Jan 1988, p. 46.



Project 14

Chaos in the Laboratory

by Mark Helmlinger

This project will allow you to use the Apple II to study the chaotic behavior of a simple mechanical system, in this case a dripping water faucet. You will build a photogate and an adjustable signal conditioner that connects to the game port. This circuit, along with the CHAOS program, allows the computer to determine the time between drops falling from a faucet. The program will then display these times in a special way that makes it easy to see the predictable structure of an apparently random series of events.

Project Requirements

Computer

This experiment can be done with the Apple II+, IIe and IIGS. The IIc, IIc+ and Laser 128 can be used if you make a minor addition to the circuit, which is described in one of the project extensions.

Background Knowledge

- Electronics: This project requires assembly of a circuit containing about a dozen components. Fairly complete instructions are given for the construction and adjustment of the photogate and signal conditioner. Previous experience with soldering is recommended.
- Mechanical skills: This project requires little mechanical assembly, except for some kind of positioning system to hold the photogate under the faucet. More advanced versions of this

experiment could require the construction or acquisition of a simple pump and reservoir system, and more sophisticated dripper nozzles.

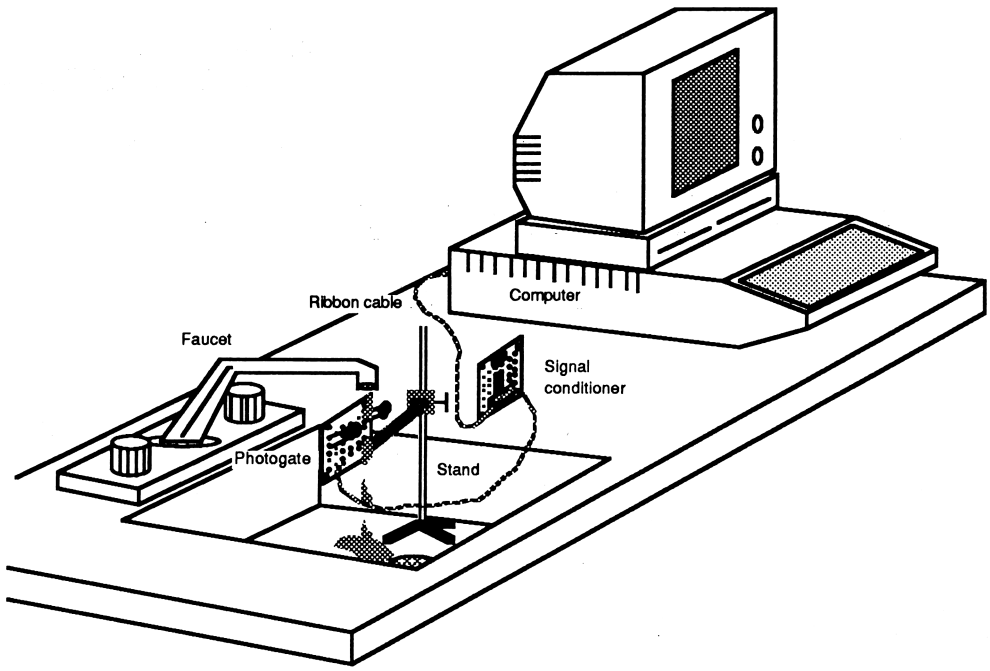
- Programming: None required for core project. Several of the project extensions require knowledge of BASIC.

Plumbing

A dripping faucet was chosen because it is an simple system, easy to measure with the Apple II, and readily available. It has been shown (see "Bibliography") that dripping faucets can display several aspects of chaotic behavior.

This project requires a sink and a faucet that drips, with a bench nearby on which to set the computer. (Remember, computers and water don't mix!) The faucet must have a valve that allows fine control of the flow. Look for drip rates of 1 to 10 drops per second. The pressure should not fluctuate too much—pressure changes can be caused by toilets, sprinklers, water fountains, or other plumbing. In addition, the water should not have too much air dissolved in it. If a suitable faucet is not available, a recirculating pump and reservoir system is described in the "Project Extensions" section of this chapter.

The figure below is an example of what the back of the lab (or your kitchen) could look like while you are learning about dripping water faucets. You may want to warn your teacher (or parents).



Chaos in the Laboratory

Parts

- ☐ circuit board (Radio Shack #276-159, we will use both halves)
- ☐ LED, any color (Radio Shack #276-041)
- ☐ infrared LED (Radio Shack #276-143 or TRW OP297A)
- ☐ infrared phototransistor (Radio Shack #276-145 or TRW OP598B)
- ☐ 1 k-Ω, 1/4 watt, carbon resistor (brown, black, red, gold) (Radio Shack #271-1321)
- ☐ 330-Ω, 1/4 watt, carbon resistor (orange, orange, brown, gold) (Radio Shack #271-1315)
- ☐ 220-Ω, 1/4 watt, carbon resistor (red, red, brown, gold) (Radio Shack #271-1313)

- ☐ 14-pin DIP socket (Radio Shack #276-1999)
- ☐ LM324 Quad Op Amp IC (Radio Shack #276-1711)
- ☐ 20-k multi-turn potentiometer (Radio Shack #271-340 or Mouser #ME323-433-20K)
- ☐ 10-μF capacitor (Radio Shack # 272-1026)
- ☐ short jumper wire (for connections on the circuit boards)
- ☐ 4-conductor cable, 1 meter long (for connecting the two circuit boards)
- ☐ cable to Apple game port (see *Appendix A*)

Note: A Chaos in the Laboratory Parts Kit, containing all of the parts listed above, is available from Vernier Software for $12.00. Two versions of this kit are available. Order Code CCH-16 includes a cable to connect to the 16-pin internal game port. Order Code CCH-9 includes a cable to connect to the 9-pin external game port. An order form is in the back of this book.

Tools and Supplies

Soldering iron, solder, wire cutters, wire strippers, needle-nose pliers, clamps, scissors, small screwdriver, and electrical tape.

Background Information

A new science of seeing
is being built at the border
between swirling randomness
and straightforward order.
Where simplicity stops
and complexity begins
lies chaos
in the space where predictability is lost
and turbulence wins.

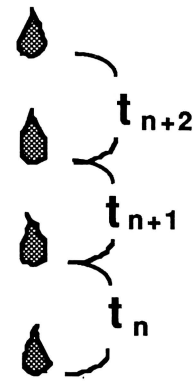
Mark Helmlinger

The purpose of this project is to examine the dynamics of a dripping faucet using a photogate connected to an Apple II computer to measure the time between drips. The series of drip times collected is then plotted in a way that makes visible certain patterns or structures that may be hidden in the data. Some patterns of drip times will be *periodic*; that is, repeating. Others will be non-periodic, but not random either. These non-repeating, non-random time patterns will fall within boundaries that have a certain structure to them. The most interesting structures that this experiment will capture and observe are called *strange attractors*. Strange indeed. Different drip rates, faucets, orifices, fluids, and temperatures will produce highly varied and interesting images of strange attractors which are characteristic of the system under investigation. Simple equations can yield data that may produce images similar to those you will obtain. The more advanced experimenter may be tempted to investigate what kind of underlying mathematical relationships could be responsible for these images.

Have you ever been kept awake at night by a dripping faucet? Some have said that a regularly dripping faucet isn't too bad, sort of like a ticking clock. Most people would agree though, that the worst are the irregular drippers. If the valve doesn't move, and the pressure behind it is constant, why does the drip pattern vary at all? Let's find a faucet and look at it closely—the one in your kitchen will do—but you may not want to let your family or friends catch you doing this. Turn it on a bit and slowly close the tap until you get a regular drip. Now, open the tap slowly and notice the drip rate increase. Pay attention to the rhythm of the drops. Do they become irregular before the flow gets too strong for separate drops? Can you distinguish any patterns? One pattern to listen for is a long time followed by a short time followed by a long time—Drip, drip - pause - drip, drip - pause - drip, drip, etc. The sequence of times between the drips has been doubled. This pattern is the result of what is called *bifurcation*, one of the paths to chaos.

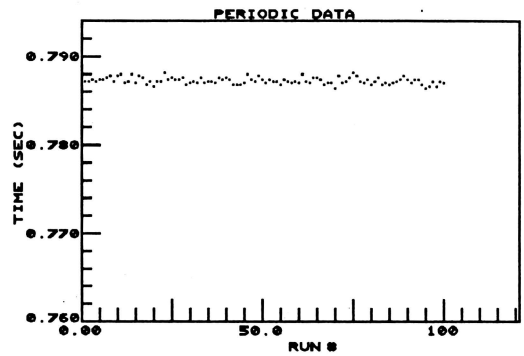
A dripping faucet or other orifice with a variable flow rate can gradually change from periodic to non-periodic drip patterns as the flow increases. This change in rhythm is an example of a system capable of something called *chaotic transition*. As the pattern changes from regular dripping to a steady stream, it may pass through regions of chaotic behavior.

For our experiment, the variable of interest is the time, t, between drops, measured in seconds. The first time collected is labeled t_1, the second t_2, and the nth drop t_n, as shown here. Any repeating pattern of times is *periodic*. The simplest case of periodic behavior occurs when $t_n = t_{n+1} = t_{n+2}$, etc., when the dripping pattern is in a *period one* mode. If the dripping is not periodic, then what you get is a series of numbers that will vary around an average value, but have no apparent pattern. For this experiment, a 1% to 20% difference in successive values can be expected.



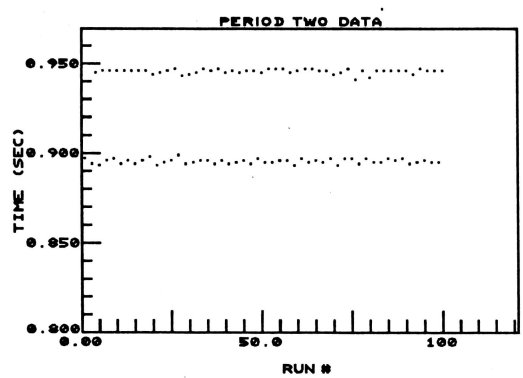
How Drop Times are Labeled

Some data sets, in fact the most interesting, will have a non-repeating pattern to them. How do you find structure in such a sequence of times, if any? One way to do this is to make a graph and to plot each drop time as a point. For a graph you need a set of Y-axis values that are related to a set of X-axis values. One way to graph the data could be to plot t_n vs. n. The graph to the right was generated from period one data. There is a bit of noise mixed in with it, which is why the line of points is not perfectly flat.



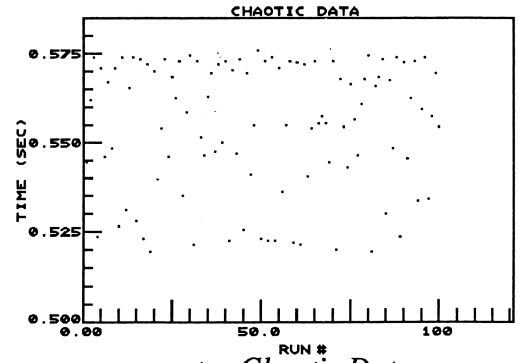
t_n vs. n, Period One Data

With a graph like that above, it would be easy to see if the drops were falling in a periodic pattern of period one (a horizontal line of dots) or not. As flow increases and the time between drops decreases, it is possible to observe a transition from a period one pattern to a *period two* pattern. This is called *bifurcation*. If a bifurcation occurs, the data will primarily have two values. The even numbered drop times will be different from the odd drop times and we would expect to see two lines of dots, as in the figure here.



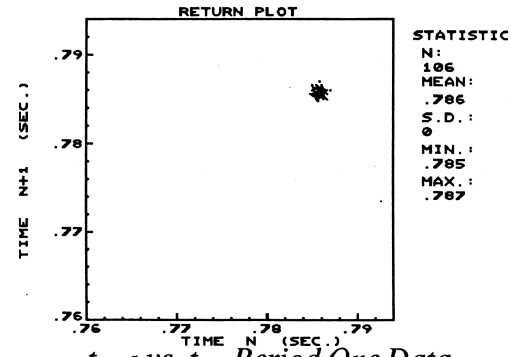
t_n vs. n, Period Two Data

However, when the system becomes chaotic, we won't be able to decipher much from these kinds of graphs. For example, this figure shows a data set that is non-repeating (non-periodic) but not totally random either. None of the times are greater than 0.575 seconds nor less then 0.515 seconds, suggesting a boundary of some kind. The data set is **not** random. The figure to the right is actually a t_n vs. n plot of an experimentally collected data set that has the structure of a strange attractor. Plotted in this way, it is difficult to distinguish chaotic from random behavior.



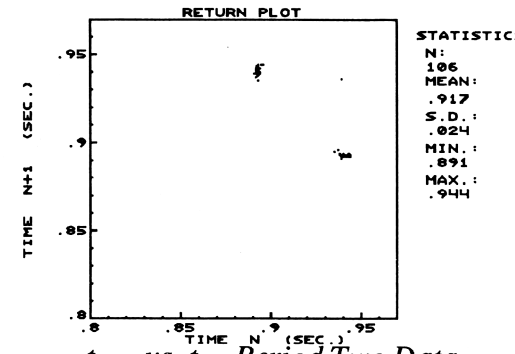
n vs. t_n, Chaotic Data

There is a better way to plot the data set of the figure above, which will tell us more about the chaotic behavior of the dripping faucet. It is called a *return plot*. An interesting method of plotting a series of numbers involves returning back to the previous value to generate coordinates for the next—hence, return plot. The time of the previous drop is used as the X-axis value for each point such that the X,Y coordinates of each point are (t_n, t_{n+1}). For example, this experiment could measure and store a series of times: 0.318, 0.306, 0.330, 0.304. The coordinates for the first four points of a return plot would then be: (0.318, 0.306), (0.306, 0.330), (0.330, 0.304), (0.304, 0.319). Eventually, if enough data is collected, a picture is built up. If $t_n \approx t_{n+1} \approx t_{n+2}$, etc., the return plot looks like this. The drops are falling in a period one pattern, plus or minus a little noise. The data used to generate this plot is the same used for the t_n vs. n graph of period one data.



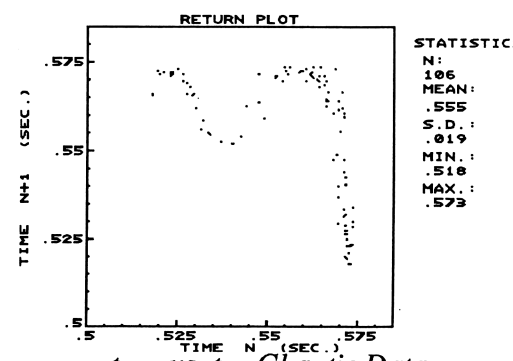
t_{n+1} vs. t_n, Period One Data

If a return plot is made from a period two data set without too much noise, it looks like this. The data used to generate this plot is the same data used for the previous t_n vs. n graph of period two data above. It would be a worthwhile mental exercise to figure out how the points were plotted, dot for dot, to obtain the figure at the right. Notice that generally, the values alternate, plus or minus a little noise: 0.892, 0.942, 0.893, 0.939, 0.891, 0.939, 0.892, 0.938, 0.892, 0.938, 0.893, 0.942, etc.



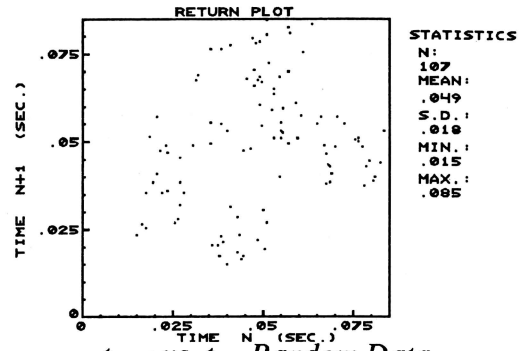
t_{n+1} vs. t_n, Period Two Data

Now, compare the previous t_n vs. n graph of chaotic data to the t_{n+1} vs. t_n plot of chaotic data here, which is the same data set plotted on a return plot. Not exactly random is it? It's not periodic either. Welcome to our chaotic world. This pattern is a strange attractor generated with data taken from a specific dripping orifice at a single, very steady flow rate. The drip times do not repeat themselves and are not periodic. Chaotic drip times do, however, wander in such a manner as to build up these interesting and sometimes enigmatic forms.



t_{n+1} vs. t_n, Chaotic Data

Let's contrast the strange attractor above to a truly random data set. If such a data series was indeed random or had a more complex chaotic structure beyond that which a return plot can point out, it might look like this. In other words, if the amount of noise in your system is too great, the pattern of dots that you get might as well have been shot from a spray paint can. If all you get are meaningless patterns like this when you do this experiment, refer to the "Troubleshooting" section.

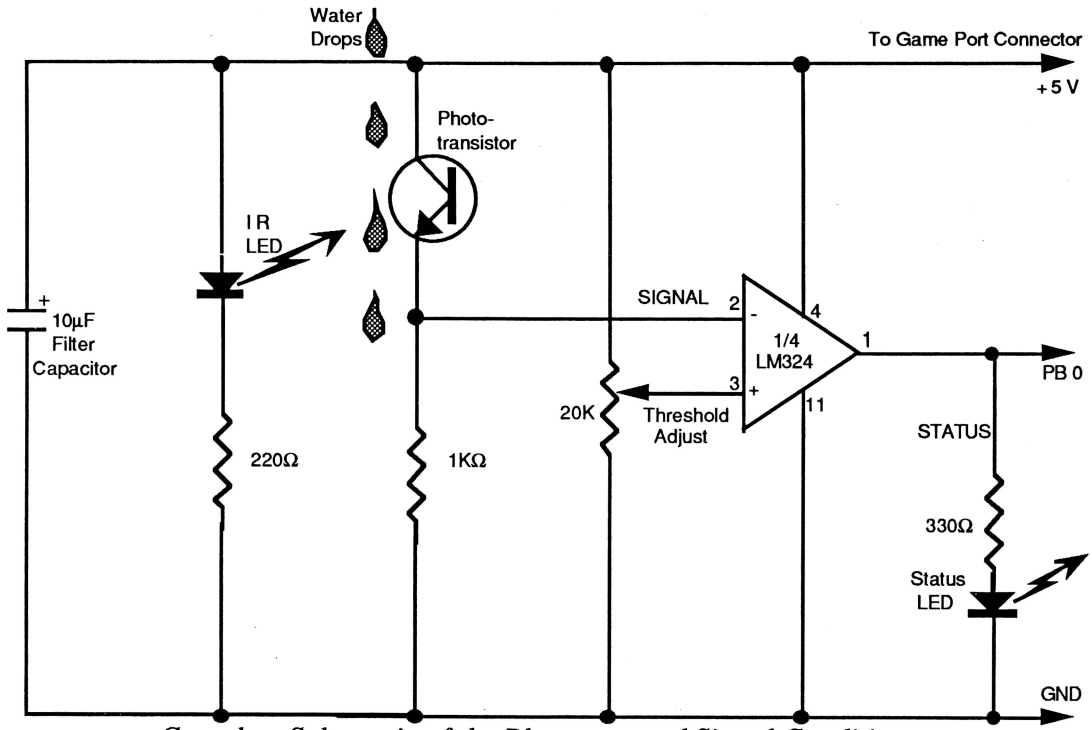


t_{n+1} vs. t_n, Random Data

More information about chaos and dripping faucets, as well as many sample return plots are included in the "Supplemental Information" section at the end of this chapter.[1] This section also includes a fairly extensive bibliography. The references cited cover many aspects of chaos—from specific mathematical explanations of the dripping water faucet, to chaos in weather, physics, chemistry, and biology.

Step-by-Step Instructions

These instructions assume that you are using a Radio Shack #276-159 circuit board. If you are using a different circuit board, the layout of parts will have to be changed. This circuit board is designed to be broken into two halves. We will use one half for the photogate, and the other half for the signal conditioner circuit. The photogate circuit board will be mounted at the dripping water stream. A 4-conductor cable will connect the photogate circuit board to the signal conditioner circuit board nearby. The signal conditioner circuit board will have one IC and a few other components on it. A cable to the Apple II game port connects to the signal conditioner circuit board. A complete schematic of the circuit is shown below:



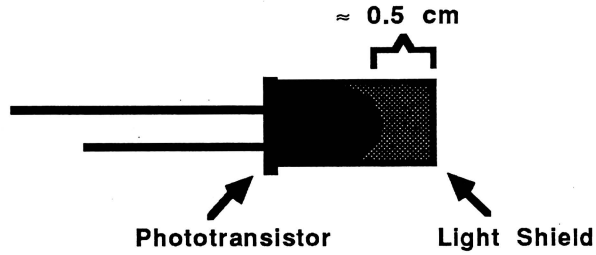
Complete Schematic of the Photogate and Signal Conditioner

[1] All of the return plots in this chapter were made using the CHAOS program from the *Chaos in the Laboratory Programs #1* disk.

Part I - The Photogate Circuit Board

☐ **1. Break circuit board into two halves.** The Radio Shack #276-159 dual circuit board is designed to allow it to be easily broken into two halves. Use one half as the photogate board and set the other one aside for use as the signal conditioner circuit board.

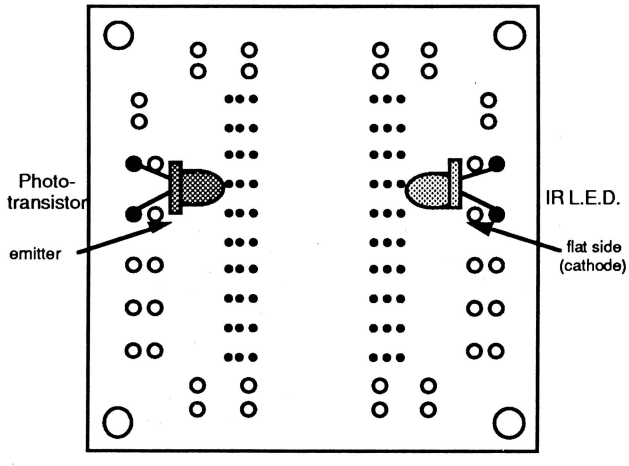
☐ **2. Prepare the infrared (IR) phototransistor.** Wrap about a 1/2" long strip of electrical tape around the phototransistor to form a shield so that stray light cannot trigger it. Trim the tape with scissors so that the phototransistor is about 0.5 cm deep. It should look like this diagram.



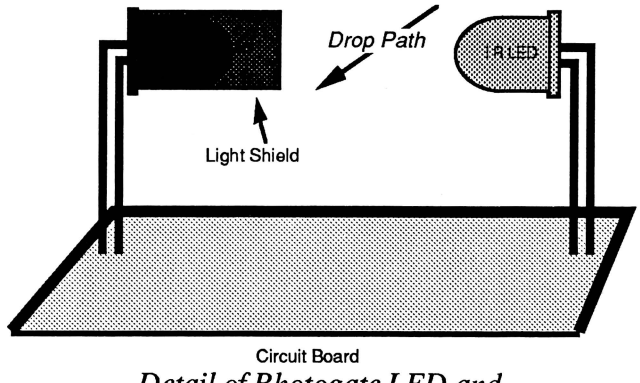
Detail of Phototransistor Light Shield

☐ **3. Mount the IR LED and phototransistor.** Insert the IR LED and phototransistor into the plain (not plated) side of the circuit board as shown below. These components are to stand up high above the circuit board. Do not cut off their leads. Pay attention to the polarity of the LED and the phototransistor. The flat side on the LED is always the negative lead. Phototransistors are marked in different ways. On many, including the one in the Vernier Software parts kit, a small bump marks the emitter. If you are using a different phototransistor, check the package carefully to determine the polarity.

The leads of these two components should be soldered to the plated side of the circuit board near their ends. Since you will be soldering a long way from the body of these components, a heat sink is not needed in this step.

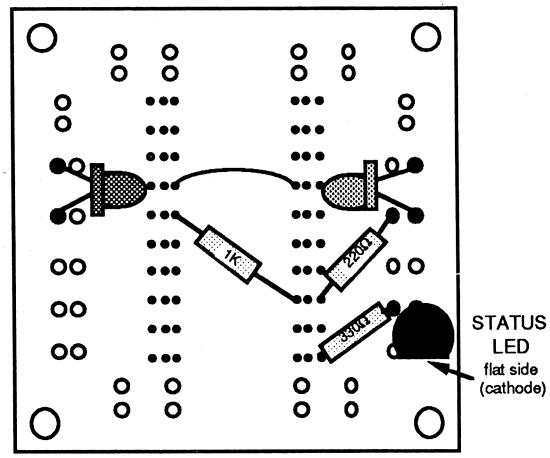


Bend the leads of the transistor and the LED so that they face one another, as shown here. The phototransistor and LED should be suspended as far above the circuit board as possible.



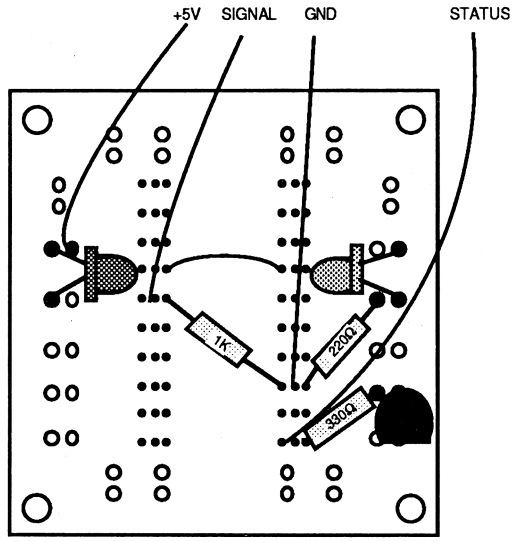
Detail of Photogate LED and Phototransistor Positioning

☐ **4. Insert and solder the other components.** Insert the leads of the resistors, LED, and jumper wire as shown below. Push them all down close to the circuit board. Bend the leads to hold them in place while soldering. Solder these all on the plated side of the circuit board and cut off the excess leads.



Photogate Circuit Components Placement

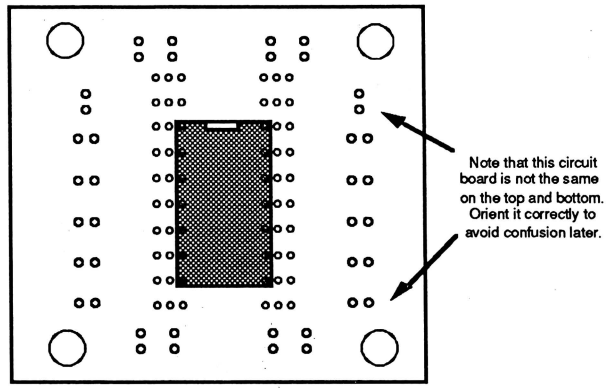
☐ **5. Connect the wires that will lead to the signal conditioning circuit.** A 4-conductor cable will connect the two circuit boards. The connections on the photogate circuit board are shown below. Separate the four wires, strip the insulation from the ends, and solder the wires in place. Label the wires "+5V", "GND", "SIGNAL", and "STATUS" so you will know how to make connections later. The wires can be threaded through one of the circuit board corner mounting holes to prevent them from breaking as they move around.



The Completed Photogate Circuit Board

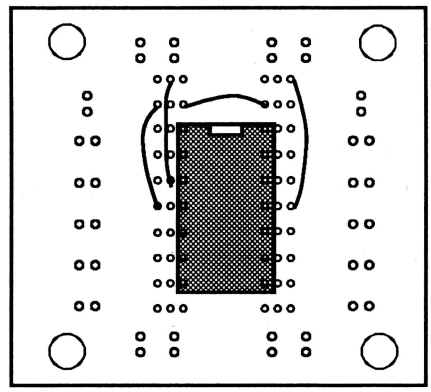
Part II - The Signal Conditioner Circuit Board

☐ **6. Solder the IC socket onto circuit board.** This part of the project uses the other half of the circuit board. Orient the 14-pin socket as shown below. Note the notch is toward the top of the circuit board. Do not insert the IC into the socket at this time. Solder all 14 pins to the plated side of the circuit board.

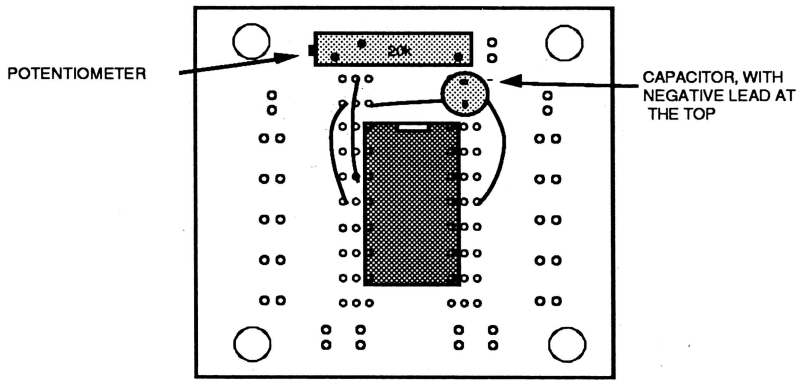


Signal Conditioner Circuit – Socket Placement

☐ **7. Connect the jumper wires.** Four short jumper wires are needed. Cut them as short as possible to keep the circuit neat. Solder them in place as shown below.

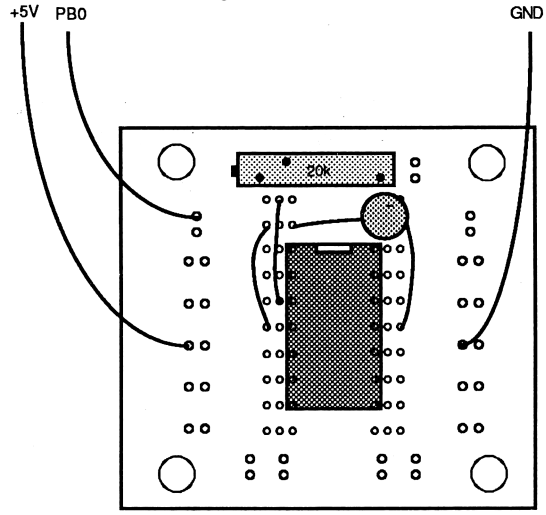


☐ **8. Add the other components.** Position the potentiometer and the capacitor as shown in this diagram. Note that the capacitor must be oriented with the lead labeled "–" toward the top of the circuit board. Its leads go into two holes right next to each other, one just above the other. The diagram shows the holes "looking through the capacitor." The three holes into which the potentiometer is placed are shown in a similar way. Solder the leads of both of these components and cut off the excess capacitor leads.



☐ **9. Prepare the cable to the Apple game port.** Refer to *Appendix A* for information on cables. Either a 9-pin cable or a 16-pin cable can be used. Only these leads will be used: +5V, PB0 and GND. Strip a few millimeters of insulation from the ends of these three leads.

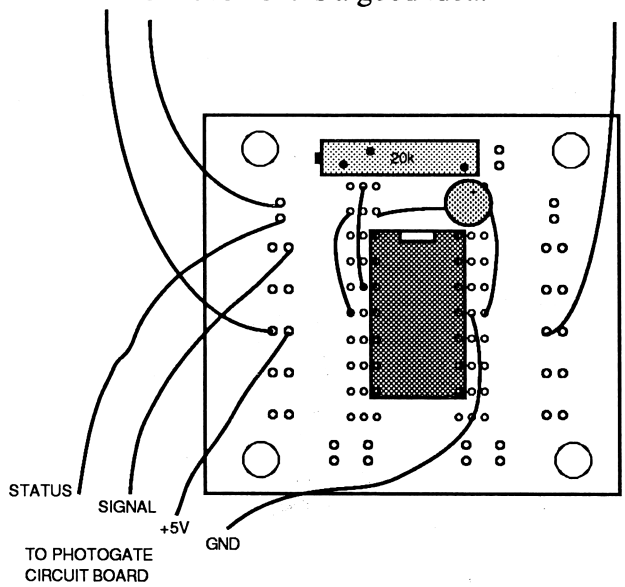
☐ **10. Connect the game port cable.** Make the three connections as shown below. You may want to thread the three leads through one of the corner holes on the circuit board to minimize movement and prevent the wires from breaking as the circuit is moved around.



☐ **11. Plug in the IC.** Carefully insert the integrated circuit into the socket. Be sure to orient pin #1 at the proper corner of the socket (to the left of the notch). Make sure all fourteen pins go into the socket. Be careful not to bend the pins too much—they can easily break.

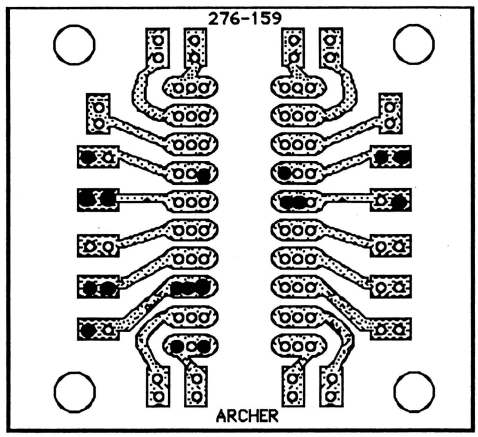
Part III - Final Assembly and Testing

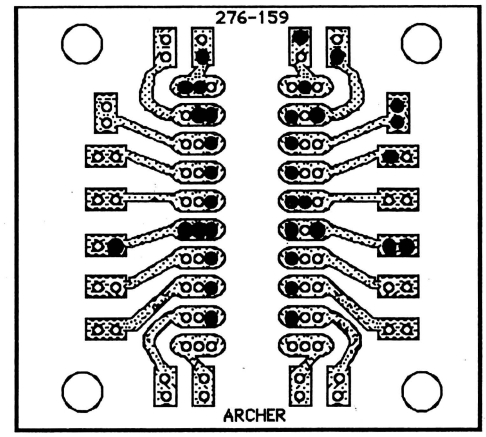
☐ **12. Connect the two circuit boards.** Solder the four leads from the photogate circuit board to the locations shown on the signal conditioner circuit board. Again, running the wires through a corner of the circuit board to minimize movement is a good idea.



Completed Signal Conditioner Circuit Board

☐ **13. Double-check all the connections.** Look over the whole circuit. Make sure everything is connected correctly. The copper-plated sides of the completed boards should look like these diagrams. The blackened holes are the only ones with wires in them. These diagrams can help you spot wiring errors you may have made. Also, check the back of the circuit boards for leads that are not soldered correctly, or solder bridges.





Photogate *Signal Conditioner*
The Backs of the Two Circuit Boards After All Components and Connections are Added

☐ **14. Test the circuit.** Turn off the computer and connect the circuit to the game port. Place the *Chaos in the Laboratory Programs #1* disk in the disk drive and turn on the computer. If the computer acts strangely or will not start, turn off the power switch and double-check your circuit—you must have an error in wiring, probably involving the +5V line.

Try moving your finger between the IR LED and the phototransistor. The status LED on the photogate board should light whenever the photogate is blocked. If not, adjust the potentiometer to a different setting using a small screwdriver. Remember that this is a multi-turn potentiometer, so you may have to turn it quite a bit before you hit the right setting. You should be able to adjust for a status LED lighted condition for different amounts of photogate blockage. In fact, that is the whole purpose of the signal conditioner.

When you get the potentiometer adjusted so that the status LED is working, run the CHAOS program. When the menu is presented, choose C - CHECK PHOTOGATE. The computer should display the status of the photogate as blocked or unblocked. It also will make noise when the photogate is blocked.

☐ **15. Find an appropriate dripper.** Test the photogate on drops of water. Position the photogate under the faucet (or whatever is dripping), making sure that the circuit board is parallel to the drip path. Use a clamp or other device to hold it in place. The photogate placement that seems to work best is about 2-3 centimeters beneath the dripping orifice, slightly off center, with the phototransistor facing away from any bright light source. Also, take care not to get the components wet while taking measurements. Brush any stray drops from the LED—they will change the focusing. Watch the status LED as you adjust the potentiometer. Try to get one pulse per drop. Use the sound produced by the CHAOS program in the C - CHECK PHOTOGATE mode to help you position the photogate and adjust the potentiometer. Listening for double drops or exceptionally long drops while making adjustments can do much to eliminate timing errors. If you are having a problem getting a single pulse for a single drop, refer to the "Troubleshooting" section. Choose the N - GET NEW READINGS mode of the program to take data for a few drops. Pressing <RETURN> will stop the data collection. A data table listing all of the times will be displayed. If every other time is very short, you are getting triggering twice per drop due to either a focusing problem or small trailing drops. Refer to the "Troubleshooting" section for suggestions on how to correct these problems.

☐ **16. Collect some data.** Adjust the faucet carefully until a periodic rhythm is heard. Take data for a few minutes. Press <RETURN> when you want to stop the data collection.[2] Notice that after you have collected some data, new options appear on the menu. The T - DISPLAY DATA TABLE

[2]If you never press a key, the program will stop data collection automatically at 1470 times.

option allows you to look at the times again. Two other new options allow you to make return plots of your data. This time, choose G - GRAPH DATA - AUTOMATIC SCALING. After you have made a return plot, some additional options will appear on the menu. These options allow you to look at or print the previous graph, or save the data on disk. The data files can also be loaded back into the CHAOS program.

Is the data really periodic? If so, increase the flow rate very slightly and observe what happens. You may find period one data, period two data, or perhaps an interesting chaotic return plot. Investigate how chaotic your dripper is at various flow rates. As the flow rate increases, it's possible that the system will go from chaotic to period one or period two patterns, instead of the other way around.

Troubleshooting

Electronic Problems

Check the following if your apparatus is not working at all:

- Is the IC plugged in with pin #1 in the right place?
- Are you sure you have used the leads from the game port correctly?
- Are you sure you have both LEDs and the phototransistor oriented the right way?
- Have you mixed up the three resistors?
- Are the IR LED and phototransistor lined up facing each other?
- Have you tried a wide range of potentiometer settings?
- If you are using an Apple IIe or IIGS, make sure you do not have anything plugged into the other game port.

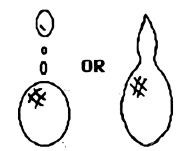
If none of this helps, try using a DC voltmeter to help you locate the problem. Check the following voltages, with the computer on and the circuit connected to the game port. For all of these checks, the negative lead of the voltmeter should be connected to the GND lead from the computer.

- Pin #4 of the IC is the positive power supply. It should measure ≈5 volts.
- Pin #11 of the IC is the negative power supply. It should measure zero volts.
- Pin #3 of the IC should be adjustable from 0 to ≈5 volts with the potentiometer.
- Pin #2 of the op-amp should be near 5 volts when unblocked and near 0 volts when blocked.
- Pin #1 is the output of the signal conditioner. It should be near 0 volts when the photogate is unblocked and over 2 volts when the photogate is blocked.
- The voltage at both the anode of the LED and the collector of the phototransistor should be +5 volts.
- The voltage at the cathode of the LED should be about 3.6 volts.

Drip Problems

Check the following if your times are not meaningful:

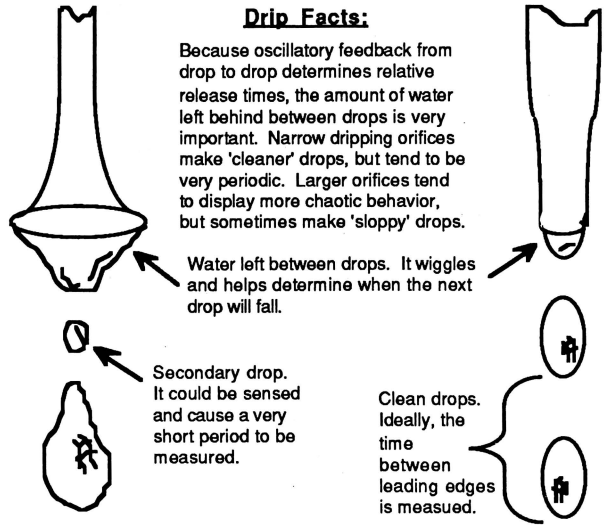
This project uses an infrared (not visible to the human eye) photogate to determine the leading edge of each drop as it falls through the sensor. The time between drops is measured, stored, and plotted. Although photogates are common in science labs these days, there is a problem with using an ordinary photogate on a water drop. A typical water drop is not always spherical.



Possible Drop Shapes

Drops of irregular shape can trigger a photogate twice per discreet drop, giving false data. The electronics part of this project describes how to build a photogate signal conditioner with an adjustable threshold that can be adjusted to trigger only on the main leading edge of the drop, and not on any of the small trailing drops that follow. Placement of the photogate relative to the falling path of the drops is also important. What can happen is that the drop can act as a lens and actually focus the infrared light from the LED into the phototransistor. This makes one drop look like two very close together.

The leading edge **and** the trailing edge of the drop block the photogate, while the middle of the drop looks like nothing is there! The solution to this problem is to carefully place the photogate so that the drop centers are just off center of the photogate beam. Adjusting the signal conditioner can help too. Be aware of the particulars pointed out in the figure below.



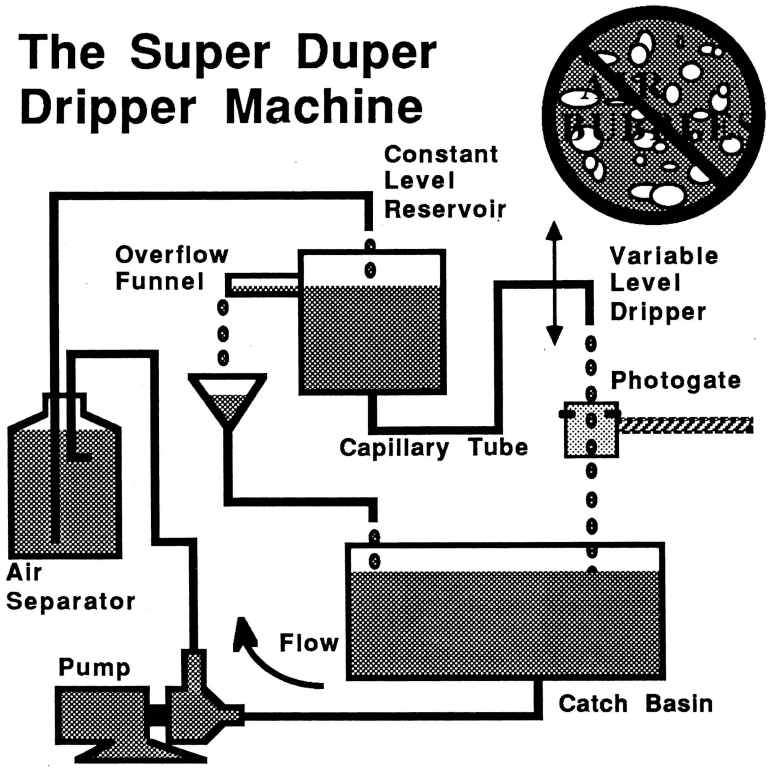
Another thing to watch out for as far as drips are concerned is air dissolved in the water. The air can come out of solution at the dripping orifice and radically change the pattern of interest. Even slight pressure changes in the water source can also change the drip pattern. If the drops are too regular and just won't vary from a period one pattern, try using a larger drip orifice. Typically, a bottle dropper from the pharmacy has too small an opening. The lip area from which the drop is suspended can be important also.

If getting one pulse per drop is difficult, connect an oscilloscope to the two ends of the 1-kΩ resistor. This will show you the signal from the photogate. Use the output of the conditioner as a trigger. The sweep will trace bumps, indicating the blocking of the beam by drops. You may see one large bump, followed by smaller ones due to secondary drips. If the big bump has two humps, you are looking at the drop refocusing the light from the LED into the phototransistor. If you are using a dual-trace scope, connect the second trace to the output of the signal conditioner and display it along with the phototransistor signal. This allows you to see exactly when the circuit is changing status.

Project Extensions

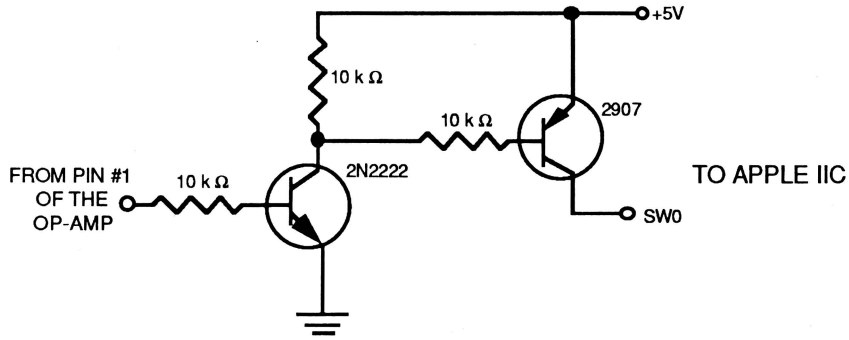
◊ Make a controlled drip rate device

If your faucets are not particularly chaotic, or if they are difficult to work with, you may want to build a dripper apparatus. The goal here is to supply a consistent and minutely controllable flow rate to a variety of dripping nozzles. The pump used should be a centripetal pump driven by a synchronous or very constant speed motor. Do not make a fountain and allow air to dissolve in the water. The air separator not only allows air time to bubble out, but also acts as a flow cushion to ensure a consistent fill rate to the reservoir. The reservoir should have a large surface area (at least 100 square centimeters) and the overflow should be large and a bit like a miniature non-turbulent waterfall to avoid backing up of the escaping water. Between the reservoir and the nozzle, turbulent flow (sharp corners, pinches, and valves) should be avoided. The capillary tube is a linear restriction to flow and exaggerates the pressure vs. nozzle height variation. This makes the system more finely adjustable. The whole apparatus should be set upon a solid bench and isolated from the vibration of the pump.



◊ Modify the circuit for use on an Apple IIc, IIc+, or Laser 128

Because the digital input circuitry of these three computers is different from the other Apple II computers, the circuit described in the "Step-by-Step Instructions" will not work with them. Add the following two-transistor circuit to provide the extra current needed to trigger the SW0 input on these computers. The CHAOS program can be used without modification.



An Apple IIc, IIc+ or Laser 128 Circuit Addition

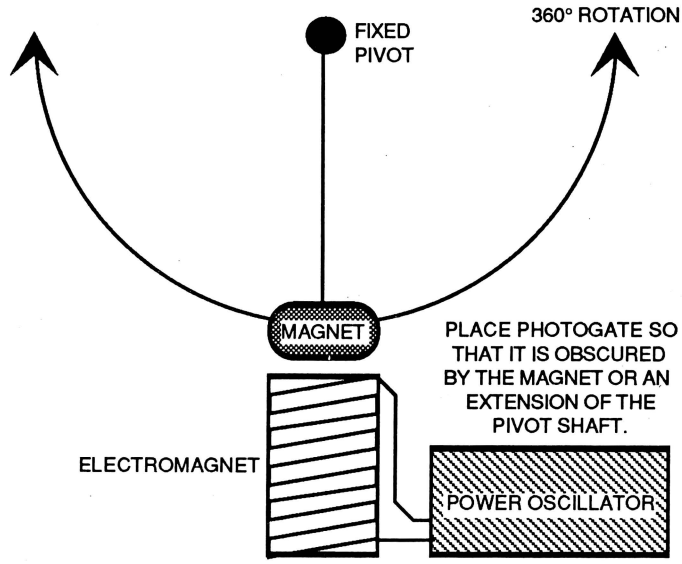
Program Improvements

The CHAOS program is an Applesoft BASIC program that can determine the count reading from the PB0 input line and convert that count into a PERIOD. The actual timing is done using a machine-language program called TIMER. These programs will not be easy for you to modify. To leave as much memory as possible for data storage, the CHAOS program has been kept fairly short. The program allows up to 1470 timings to be made. It also has been compressed using the Beagle Bros. program COMPACT. This compressing removes all REM statements and combines lines to save as many bytes as possible. It also makes the program difficult to read. If you increase the size of the CHAOS program very much, you will start to have problems with insufficient memory. As you modify the program, either keep your changes very short or delete sections of the program as you add new ones. Here are some modifications you might want to try:

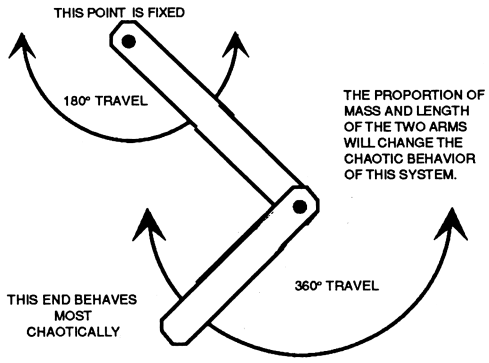
- Change the digital input line used. The CHAOS program uses the PB0 input line of the Apple II game port. If, for any reason, you want to connect the photogate to a different input line (PB1 or PB2), change line 2630 of the CHAOS program.
- The program does not support printing out data tables. Add the appropriate statements in line numbers 8000-8999 to turn on and off the printer to support printing of the data.
- You may want to BSAVE the graphs produced by this program so that you can use them in other Apple II graphing programs. If you would like to save graphs on disk frequently, add the appropriate commands to the main menu of the CHAOS program.
- Data saved by the CHAOS program is stored in a simple ASCII text format, so you may want to load it into other programs or even other computers for further analysis. The data files can also be loaded back into the CHAOS program.

Additional Experiments

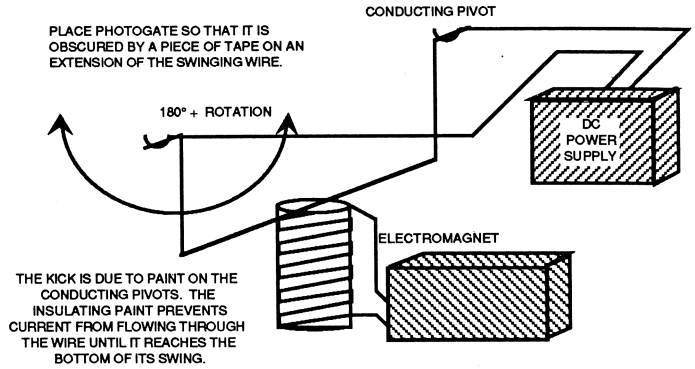
The photogate and computer apparatus described in this chapter can be used to investigate the chaotic behavior of a variety of physical systems. One of the easiest systems to model that is capable of chaotic motion is the *driven pendulum*. The driven pendulum is a swinging mass that is regularly pushed at a frequency different from its natural frequency. There are several variations on this theme, including the kick driven pendulum, the double pendulum, and the dip needle. For all simple systems, there is a driving frequency, a restoring force, and damping. Another example of chaotic motion often studied is of a "flag" flapping in the breeze. Shown here are diagrams of the physical systems described. The book by Moon (see "Bibliography") describes these and many other simple physical systems this project that can be investigated using the photogate system from this project. Try some of these or ideas you come up with yourself. Will strange attractors from a "complex" system be similar to strange attractors from a "simple" system?



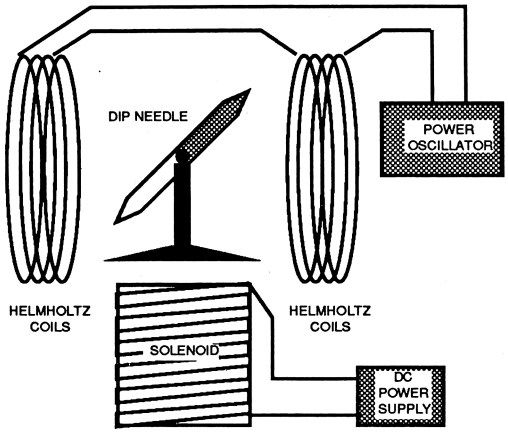
The Driven Pendulum



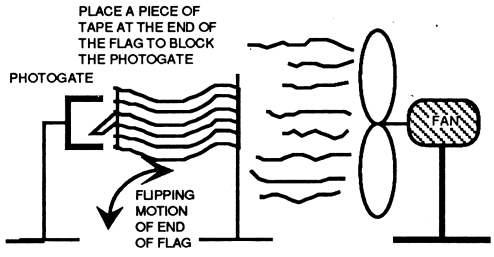
The Double Pendulum



The Kick Driven Pendulum



The Chaotic Compass Needle



The Flapping Flag

Supplemental Information

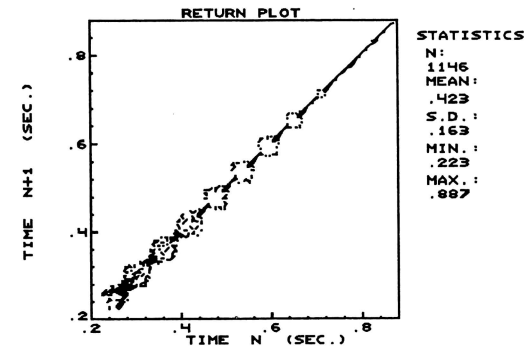
More about Chaos

Let's talk about the weather. One may wonder why weather predictions never go very far into the future. Chaos theory was in part developed by a meteorologist named Edward Lorenz. In the early sixties he was working on a computerized model for predicting long-term global weather patterns. As he refined his model, he soon discovered that long-term weather prediction was utterly impossible. His model was so sensitive to initial conditions that the wind from the wings of a butterfly in New York could change next week's weather in Los Angeles! At the same time, other scientists were finding that deterministic (predictable) systems could lead to complex apparently random behavior. A century earlier, H. Poincaré showed that uncomplicated systems governed by regular and well-known laws can be extremely sensitive to initial conditions. Infinitely small measurement errors in initial values can grow with time in such a way as to make the future state of an apparently simple system totally unpredictable. This hypersensitivity to initial conditions seems to be an essential quality of chaotic behavior, and is actually called the *Butterfly Effect*. This and other ideas are explained and illustrated in an interesting NOVA program called *The Strange New Science of Chaos*.

Chaos can mask a hidden simplicity. We know from everyday experience that although the weather may be somewhat unpredictable, it's not completely random either. Generally, it doesn't snow in July, and heat prostration isn't a problem in January. There seems to be boundaries and patterns to the disuniformities of nature, and chaos is a tool that can be used to see them. It is the shape of these boundaries of predictability and whatever structures lie within that is the study of chaos.

More About Dripping

You might be tempted at this point to wonder in what order period one vs. period two vs. chaotic data occur when the flow rate is varied. To answer that question, and a few others, data was collected while the flow rate was slowly changed, generating this plot.



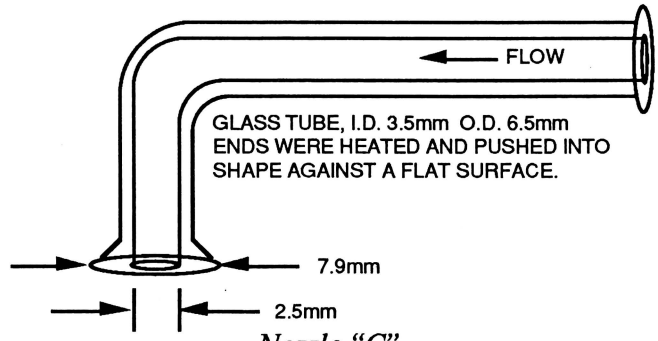
Map of Behavior Over Range of Flow Rates

This is an overall map of the system's behavior over a wide range of flow rates. It was made by collecting data while the flow rate was gradually decreased. It is possible to generate this map if you are using a drip system which has a reservoir that you can allow to empty slowly. It can give you an idea of what general kind of dripping pattern to expect for the flow rate you'd like to concentrate on.

Let's say you would like to collect data from your apparatus at a flow rate that produces drops around 0.6 seconds apart. For the dripper that made the map above, we can see that at about 0.6 seconds, the drops are dripping in a period two pattern (two different times dominate). A little more flow, or a faster drip rate, will cause a period one pattern. An even faster flow rate will cause another period two pattern, but perhaps with a little more complex structure. Taking detailed data at just one flow rate will give the system enough time to generate a meaningful return plot. Perhaps 1,000 points or more could be collected, and most will overlap somewhat. This can build up a more easily recognizable picture.

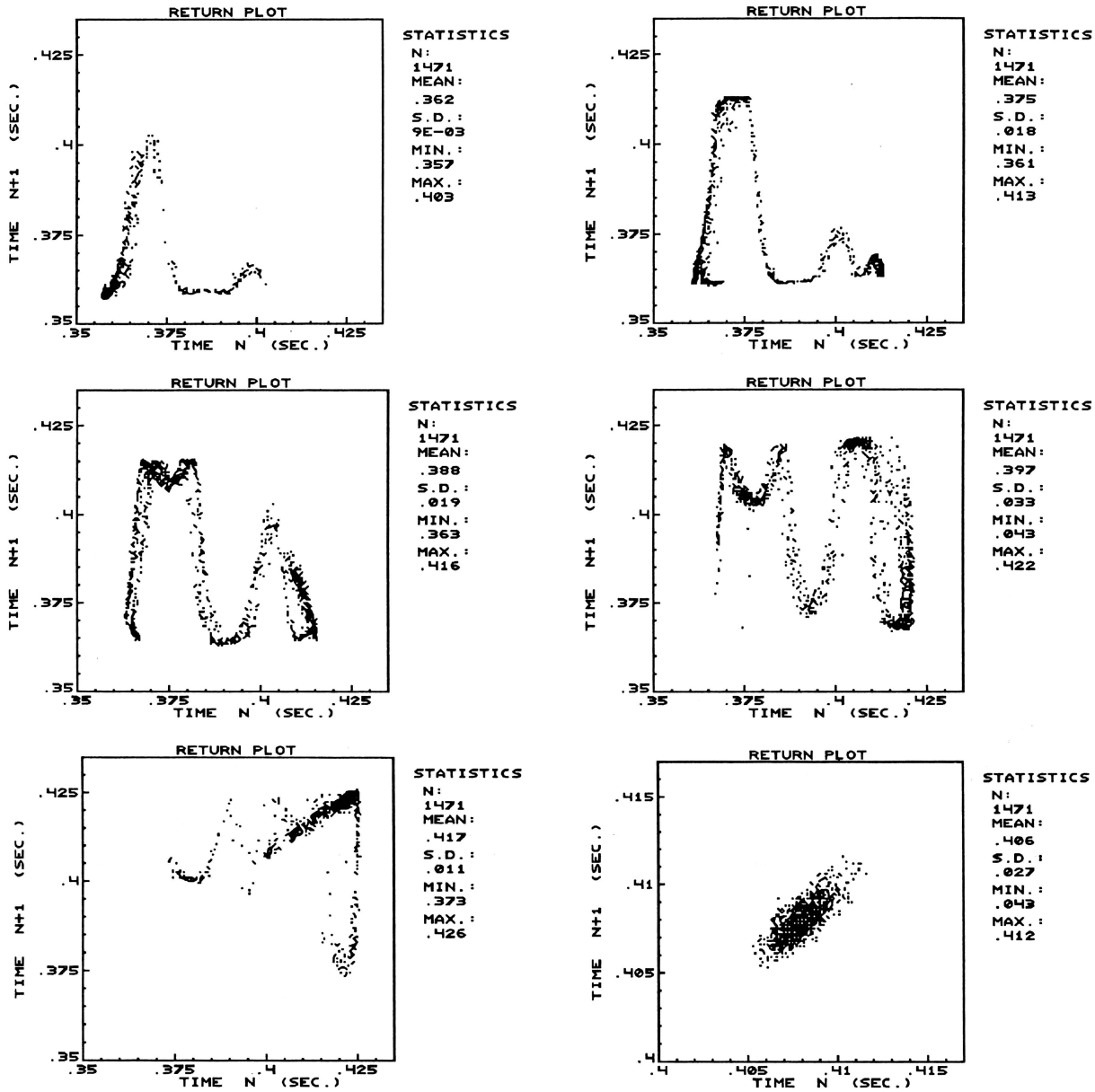
The map above implies that for the system used, there are many "windows" of period one behavior. Period one dripping could be found at almost any drip rate, depending on the dripper. This could be confusing if you're expecting the complexity of the drip pattern only to increase with flow. Regions of interesting behavior may not necessarily be at the higher flow rates for your system. For example, you could expect water flowing out of a hose to go from smooth to chaotic as the flow rate is increased. However, the dripping faucet is more complex because of the way the water bounces around in the nozzle after a drip has left. This wiggling, or oscillation, allows one drop to communicate to the next, and help determine when the next drop will fall. That's one reason why we use a return plot to compare the time of one drop with the next. The relationship between the rate at which a drop is filled and its frequency of oscillation determines if the wiggling will make it drip sooner or later or at the same time as the last.

Part of the research for this chapter involved forming various dripping nozzles from a glass tube and making detailed maps of each. The map above was made using nozzle "C". The physical description of nozzle "C" appears below:



Nozzle "C"

The following series of images was made from data collected with nozzle "C". Note that each is a strange attractor for a specific flow rate. The relative flow rate of the system is related to the mean time between drops. The flow rate was decreased between each state. Note the evolution of patterns from chaotic to period one. All plots, except the one on the bottom right, are plotted on the same scale for comparison. The last plot is "blown up" to show the lack of structure, even on the fine scale.



Return Plots all Made Using Nozzle "C," with Slightly Different Flow Rates

Bibliography

Beginner's Reading List (Books):

Briggs, John. *Turbulent Mirror: An Illustrated Guide to Chaos Theory and the Science of Wholeness.* New York: Harper and Row, 1989.

Gleick, James. *Chaos: Making a New Science.* London: Heinemann, 1988.

Stewart, Ian. *Does God Play Dice? The Mathematics of Chaos.* New York: B. Blackwell, 1989.

Beginner's Reading List (Articles):

Arecchi, Tito. "Chaos and Complexity." *Times Literary Supplement*, 6 Oct. 1989, p. L16.
Begley, S. "Finding the Order in Chaos." *Newsweek*, July 18, 1983, p. 53.
Begley, S. "Finding the Order in Disorder." *Newsweek*. 21 Dec. 1987, pp. 55-56.
Brady, Diane. "The Roots of Chaos: An Exciting Theory Stirs the Scientific World." *MacLeans*, 19 Nov. 1990, pp. 46-47.
Coveney, Peter. "Chaos, Entropy and the Arrow of Time." *New Scientist*, 29, Sept. 1990, p. 49.
Crutchfield, J.P., Farmer, J.D., Packard, N.H., and Shaw, R.S. "Chaos." *Scientific American*, Dec. 1986, pp. 46-57.
Davies, Paul. "Chaos Frees the Universe." *New Scientist*, 6 Oct. 1990, p. 48.
Gibbons, A. "Chaos and The Real World." *Technology Review*, July 1988, pp. 10-11.
Gleick, James. "New Images of Chaos that are Stirring a Science Revolution." *Smithsonian,* Dec. 1987, pp. 4+.
Gleick, James. "Solving the Mathematical Riddle of Chaos." *New York Times Magazine*, 10 June 1984, pp. 30-32+.
Horgan, J. "Nonlinear Thinking." *Scientific American*, June 1989, pp. 26+.
Lees, D. "A Science Called Chaos." *MacLeans*, 1 Oct. 1984, pp. 76-77+.
Mandelbrot, Benoit. "Fractals - A Geometry of Nature." *New Scientist*, 15 Sept. 1990, p. 38.
May, Robert. "The Chaotic Rhythms of Life." *New Scientist*, 18 Nov. 1989, p. 37.
McRobie, Allan, and Thompson, Michael. "Chaos, Catastrophes and Engineering." *New Scientist*, 9 June 1990, p. 41.
Mullin, Tom. "Turbulent Times For Fluids." *New Scientist*, 11 Nov. 1989, p. 52.
Murray, Carl. "Is the Solar System Stable?" *New Scientist*, 25 Nov. 1989, p. 60.
"Order Into Chaos. "*New Scientist,* 1 Oct. 1989, p. 24.
Palmer, Tim. "A Weather Eye on Unpredictability." *New Scientist*, 11 Nov. 1989, p. 56.
Percival, Ian. "Chaos: A Science for the Real World." *New Scientist*, 21 Oct. 1989, p. 42.
Scott, Stephan. "Clocks and Chaos in Chemistry." *New Scientist*, 2 Dec. 1989, p. 53.
Series, Caroline. "Fractals, Reflections, and Distortions." *New Scientist*, 22 Sept. 1990, p. 54.
Stewart, Ian. "Portraits of Chaos." *New Scientist*, 4 Nov. 1989, pp. 42.
Taubes, G. "The Mathematics of Chaos." *Discover*, Sept. 1984, pp. 30-32+.
"The Images of Chaos." *Discover*, Sept. 1984, pp. 40-45.

Intermediate Reading List (Books):

Baker, Gregory L. *Chaotic Dynamics: An Introduction*. New York: Cambridge Univ. Press, 1990.
Devaney, Robert L. *Chaotic Dynamical Systems*. Redwood City, CA: Addison Welsey, 1989.
Moon, Francis C. *Chaotic Vibrations*. New York: John Wiley & Sons, 1987.
Shaw, Robert. *The Dripping Faucet as a Model Chaotic System*. Santa Cruz, CA: Aerial Press, 1984.

Intermediate Reading List (Articles):

Bishop, Alan R. "Simplifying Complexity." *Nature*, 22 March 1990, pp. 290-291.
Cahalan, Robert F. and Others. "Chaotic Rhythms of a Dripping Faucet." *Computers in Physics*, July/Aug. 1990, pp. 368-382.
Chernikov, A.A. et. al. "Chaos: How Regular Can it Be?" *Physics Today*, Nov. 1988, pp. 27-35.
Cipra, B.A. "Computer-Drawn Pictures Stalk the Wild Trajectory." *Science*, 2 Sept. 1988, pp. 1162-1163.
Crutchfield, James P. "Quantifying Spatial Chaos." *Research and Developmen*t, 21 Nov. 1989, pp. 12-17.
Devaney, R.L. "Chaotic Bursts in Nonlinear Dynamical Systems." *Science*, 16 Jan. 1987, pp. 242-245.
Dewdney, A.K. "Probing the Strange Attractions of Chaos." *Scientific American*, July 1987, pp. 108-111.
Gell-Mann, Murray. "Simplicity and Complexity in the Description of Nature." *Engineering and Science*, Spring 1988, pp. 3-9.

Grebogi, C. and Others. "Chaos, Strange Attractors, and Fractal Basin Boundaries in Nonlinear Dynamics." *Science*, 30 Oct. 1987, pp. 632-638.
Kadanoff, L.P. "Roads to Chaos." *Physics Today*, Dec. 1983, pp. 46-53.
Maddox, John. "Is Chaos Becoming Conversational?" *Nature*, 7 Sept. 1989, pp. 17-18.
Martien, P., Pope, S.C., Scott, P.L., and Shaw, R.S. "The Chaotic Behavior of the Leaky Faucet." *Physics Letters*, 110A (1985): 399-404.
May, R.M. "Simple Mathematical Models With Very Complicated Dynamics." *Nature* 261 (1976): 459-467.
Nelson, A.R. and Gokhale N.R. "Oscillation Frequencies of Freely Suspended Water Drops." *Journal of Geophysical Research* 77 (1972): 2724-2727.
Peterson, I. "Beyond Chaos: Ultimate Unpredictability." *Science News*, 26 May 1990, p. 327.
Peterson, I. "Escape Into Chaos." *Science News*, 26 May 1984, pp. 328-329.
Peterson, I. "In the Shadows of Chaos." *Science News*, 3 Dec. 1988, pp. 360-361.
Peterson, I. "Zeroing in on Chaos." *Science News*, 28 Feb. 1987, pp. 137-139.
Pool, R. "Where Strange Attractors Lurk." *Science*, 10 March 1989, p. 1292.
Rothman, T. "God Takes A Nap." *Scientific American*, Oct. 1988, p. 20.
"The Amateur Scientist, Why a Fluid Flows Faster When the Tube is Pinched." *Scientific American*, July 1987, pp. 104+.
"Unmashing Chaos." *Scientific American*, Feb. 1987, pp. 61-62.

Advanced Reading List (Books):

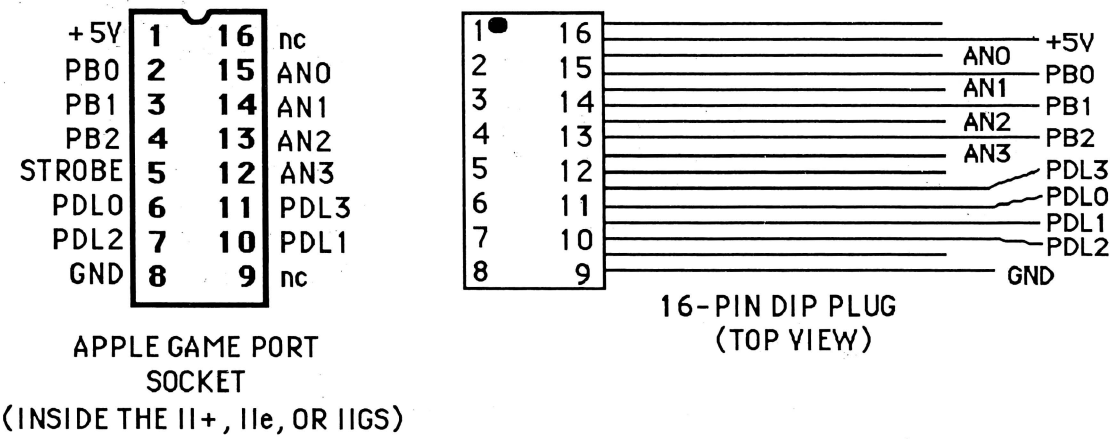
Campbell, David K. *Chaos/XAOC: Soviet-American Perspectives on Nonlinear Science*. New York: American Institute of Physics, 1990.
Holden, Arun V. *Chaos*. Princeton, NJ: Princeton University Press, 1986.
Lundquist, Stig. *Order and Chaos in Nonlinear Physical Systems*. New York: Plenum Press, 1988.
Mandelbrot, B.B. *The Fractal Geometry of Nature*. San Francisco: W.H. Freeman and Co., 1983.
Rasband, S. Neil. *Chaotic Dynamics of Nonlinear Systems*. New York: Wiley, 1990.

Appendix A
Cables to the Apple Game Ports

If you are using an Apple II+, you need to use a 16-conductor ribbon cable to the internal game port. If you are using an Apple IIe or IIGS, you have a choice of two different cables to use: a 16-conductor cable to the internal game port or a 9-conductor cable to the external game port. If you are using an Apple IIc, IIc+, or Laser 128 you will need a cable to a 9-pin game port. These two types of cables are described below:

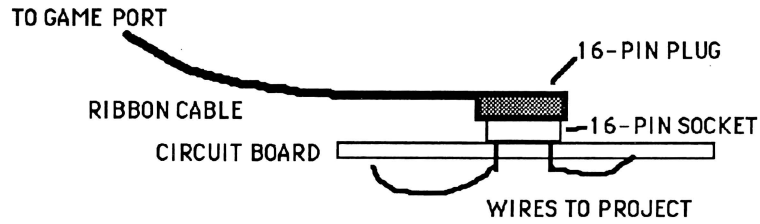
16-Conductor Cable to the Internal Game Port (II+, IIe, or IIGS)

Most Apple II computers (except the IIc and IIc+) have a 16-pin DIP (**D**ual **I**n-line **P**in) connector inside the computer for use as a game port. To use this connector, you have to remove the top cover of the computer.[1] The connector looks like an IC socket. The pins and a ribbon cable with a plug to connect to the connector are shown below. Notice that there is a notch or a mark on the circuit board to indicate pin #1 of the game port.



Apple II+, IIe, IIGS Internal Game Port

16-conductor ribbon cables with DIP plugs are sold at many electronic-parts supply houses. Digi-Key Corporation sells 16-pin ribbon cables in many different configurations. Their part #C2PXT-1636M-ND and #A2PXT-1636M-ND are both 36" long and have a 16-pin DIP connector on one end. They sell for about $3 to $5. You can solder the appropriate leads on the other end to your project. An alternative is to purchase a cable with a DIP plug on each end. This allows you to use the same cable for several projects. You can insert one DIP connector into a 16-pin DIP IC socket (Radio Shack #276-1998) mounted on your project. Connect the wires from the pins of this socket to the project as shown below:

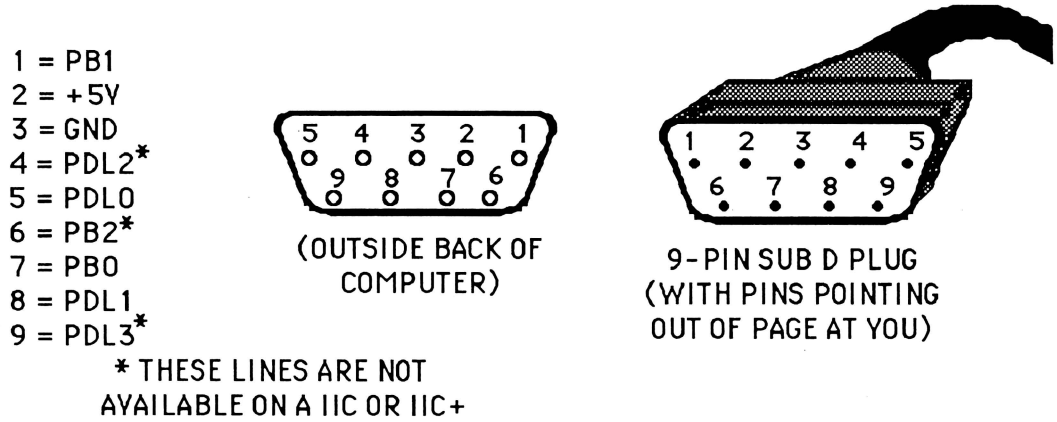


[1]That is, unless you are using a "game port extender," which acts as an extension cord to make this socket accessible from outside the computer. This makes using the 16-pin game port much easier. Devices of this type are sold by a number of companies, including Vernier Software (Order Code GPE, $16.00).

When using a 16-conductor ribbon cable, mark one corner of the plug as the #1 pin and always match this pin with the notch on the connector. It is easy to plug a 16-pin DIP plug in the wrong way. Another potential problem with this kind of connector is bending the pins. The pins are made of very fine wire and can easily bend as you plug them in. Once they bend, they often break when you straighten them out. Worse yet, sometimes you do not notice that the pins have bent instead of making connection, and your circuit will not work. To avoid these problems, make sure all of the pins are straight and in line as you plug in the connector.

9-Conductor Cable to the External Game Port (IIe, IIc, IIc+ or IIGS)

All Apple II computers except the Apple II+ have a game port located on the outside back of the computer. It is somewhat more convenient to use, because you do not have to take the top off the computer to reach it. This game port uses a 9-connector sub-D socket. The annunciator-output lines AN0, AN1, AN2, and AN3 are not present on this connector. This means the 9-conductor cable **cannot** be used for any project that involves controlling electrical devices. Specifically, this connector cannot be used for the following projects in this book: Ultrasonic Range Finder, Stepper Motors, A-to-D Converter, Colorimeter, D-to-A Converter/Digitized Sound, and Smart Battery Charger.



Apple IIe, IIGS, IIc, IIc+ External Game Port

The plug that goes to this game port is a 9-conductor sub-D plug, which is available at Radio Shack (#276-1537) and most other electronic-parts supply houses. Solder either a ribbon cable or a multiple-conductor round cable to this connector. If you do not want to spend time soldering leads onto the plug, purchase an assembled 9-cable with a plug on the end from Digi-Key Corporation. Part #C7MXT-0906M-ND or #A7MXG-0906G-ND is six feet long and sells for about $5 to $7. Radio Shack sells a Joystick Extension Cable (Radio Shack #270-1705). This is a 10-foot long cable with a sub-D plug (pins sticking out) on one end and a sub-D socket (with holes) on the other end. You can use the plug end to connect to the Apple II. Cut off the other end and set it aside. Strip the wires on the cut end of the plug cable for soldering to your project. You will need to use an ohmmeter to determine which wire connects with each pin.

Appendix B
Resistor Color Codes

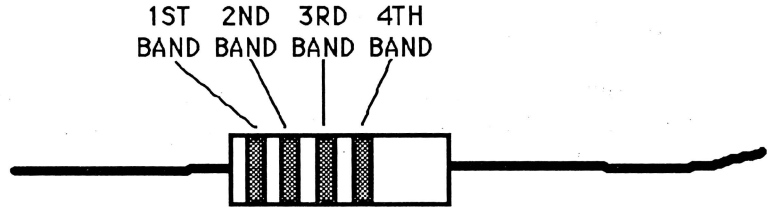
Almost all resistors are labeled using the following colors to represent numbers:

black	0	blue	6
brown	1	violet	7
red	2	gray	8
orange	3	white	9
yellow	4	gold	0.1
green	5	silver	0.01

The labeling system used is slightly different for carbon-composition resistors and metal-film resistors.

Carbon-Composition Resistors

Most ordinary carbon-composition resistors are labeled with four color bands. The bands are read in order starting at the end of the resistor with a color band nearest it. The figure below illustrates this:



Carbon Composition Resistor

The first two bands are read as a two-digit number. The third band is read as a power of 10 multiplier of that number.

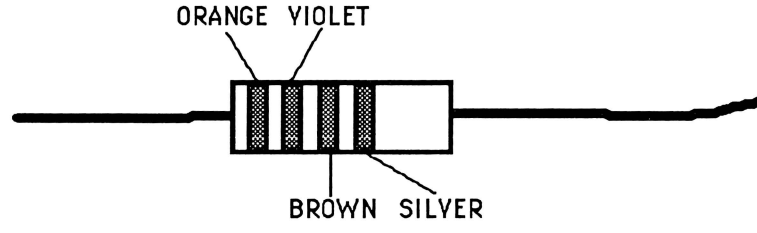
resistance value = (2-digit # from first 2 bands) x $10^{\text{(third band digit)}}$

The fourth band is an indicator of the tolerance of the resistor. The tolerance is stated as a percent of the labeled value, which is the most that the actual resistance can differ from the labeled value. A special color code is used for the tolerance color code:

no color	20%	gold	5%
silver	10%	brown	1%

Consider some examples:

Example 1:

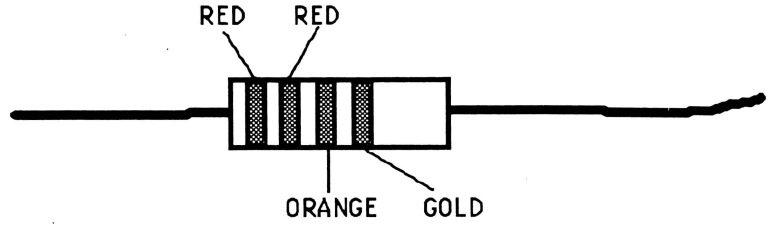


The value is:

$$37 \times 10^1 = 37 \times 10 = 370\ \Omega$$

The tolerance is plus or minus 10%.

Example 2:



The value is:

$$22 \times 10^3 = 22 \times 1000 = 22{,}000\ \Omega \text{ or } 22\ k\Omega$$

The tolerance is plus or minus 5%.

Metal-Film (Precision) Resistors

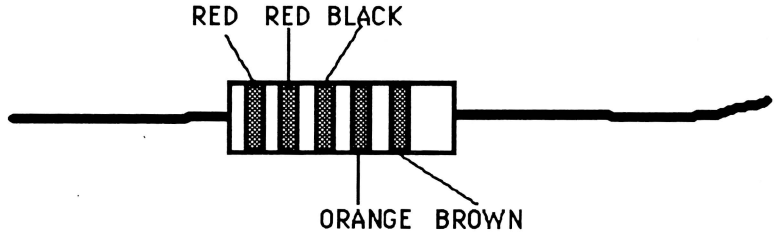
Metal-film resistors are labeled slightly differently. They are marked with five color bands. The first three bands indicate a three-digit number. The fourth band is a power of 10 multiplier of that number.

$$\text{resistance value} = (\text{3-digit \# from first 3 bands}) \times 10^{(\text{fourth band digit})}$$

The fifth band is used to represent the tolerance of the resistor. It is usually brown because most metal-film resistors have a tolerance of 1%.

Sometimes it is difficult to determine which end of the resistor to begin with when using metal-film resistors. Remember that the fifth band on a metal-film resistor is usually brown. If all else fails, you can check the resistance with an ohmmeter.

Here is an example of reading a metal-film resistor's value from the color bands:



Metal Film Resistor

The value is:

$$220 \times 10^3 = 220 \times 1000 = 220{,}000\ \Omega \text{ or } 220\ k\Omega$$

The tolerance is plus or minus 1%.

Appendix C
Bibliography

Electronics

Barnaal, Dennis. *Digital and Microprocessor Electronics for Scientific Applications*. North Scituate, MA: Breton Publishers, 1982.

Horowitz, Paul and Winfield Hill. *The Art of Electronics*. Cambridge: Cambridge University Press, 1988.

Malmstadt, Howard V., C.G. Enke and S.B. Crouch. *Electronics and Instrumentation for Scientists*. New York: Benjamin-Cummings, 1982.

McComb, Gordon. *The Robot Builder's Bonanza*. Blue Ridge Summit, Pennsylvania: TAB Books, 1987.

Meiksin, Z. H. and Philip C. Thackray. *Electronic Design with Off-the-Shelf Integrated Circuits*. West Nyack, New York: Parker Publishing Co., 1980.

Programming

De Jong, Marvin L. *Apple II Assembly Language*. Indianapolis: Howard W. Sams, 1982.

De Jong, Marvin L. *Programming and Interfacing the 6502, with Experiments*. Indianapolis: Howard W. Sams, 1980.

Wagner, Roger. *Assembly Lines: The Book*. North Hollywood, CA: Softalk, 1982.

Laboratory Interfacing

Coffron, James W. *The Apple Connection*. Berkeley: SYBEX, 1982.

Titus, Jonathan A., David G. Larsen and C.A. Titus. *Apple Interfacing*. Indianapolis: Howard W. Sams, 1981.

Engelsher, Charles J. *Interfacing and Digital Experiments with Your Apple*. Blue Ridge Summit, Pennsylvania: TAB Books, 1984.

De Jong, Marvin L. *Apple II Applications*. Indianapolis: Howard W. Sams, 1983.

Hallgren, Richard C. *Interface Projects for the Apple II*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1982.

Science

Weast, Robert C., ed. *Handbook of Chemistry and Physics*, 66th Edition. Boca Raton, FL: The Chemical Rubber Co., 1985.

Appendix D

Software on the *Chaos in the Laboratory Programs* Disks

The files used by each of the projects are listed below. In some cases the same file is used for more than one project. Each file is identified by type and a brief description is provided. The assembly-language, source-code files were all made using the ProDOS Editor/Assembler.

Project 1. Thermistor (on /CHAOS1)

THERMISTOR	BASIC	displays the temperature read by the thermistor
THERMISTOR.2	BASIC	displays the temperature read by two thermistors
PDL.TESTER	BASIC	simple program to display PDL line status
THERM.CAL	TXT	calibration file for the thermistor
THERM.CAL.2	TXT	calibration file for two thermistors
PDL.CHECK	MACHINE	determines the resistance as a count
PDL.CHECK.SRC	ASSEMBLER	source code for PDL.CHECK

Project 2. Quiz Show (on /CHAOS1)

QUIZ	BASIC	Quiz Show program to monitor buttons
PB.TESTER	BASIC	simple program to display the status of each PB line
LP.1200	MACHINE	routine to display large letters on the screen

Project 3. Computer Vision (on /CHAOS1)

VISION	BASIC	main Computer Vision program
PDL.TESTER	BASIC	reports the status of each PDL line

Project 4. Colorimeter (on /CHAOS1)

COLOR	BASIC	Colorimeter program
COLOR.TESTER	BASIC	turns on each LED in sequence for testing
COLOR.USR	BASIC	colorimeter program for use with low light levels
XP	MACHINE	routine used by COLOR.USR to read PDL lines

Project 5. A-to-D Converter (on /CHAOS2)

A.TO.D	BASIC	program to read and display voltage
A.TO.D.SRC	ASSEMBLER	source code for A.TO.D.0
A.TO.D.0	MACHINE	routine to read the voltage from the serial A-to-D Converter (relocatable)
A.TO.D.AVG	BASIC	program to read voltage 64 times and report the average reading
A.TO.D.AVG.SRC	ASSEMBLER	source code for A.TO.D.AVG.0
A.TO.D.AVG.0	MACHINE	routine used by A.TO.D.AVG for the actual voltage reading
SCOPE	BASIC	program to simulate an oscilloscope
OSC.0	MACHINE	machine code for the oscilloscope
SCOPE.SCREEN	BINARY	background screen for oscilloscope
OSC.DATA	BINARY	look-up table for quickly plotting points on the screen

Project 6. Light Meter (on /CHAOS2)

LIGHT	BASIC	Light Meter program
A.TO.D.AVG.0	MACHINE	routine to read the A-to-D Converter (relocatable)
A.TO.D.AVG.SRC	ASSEMBLER	source code for A.TO.D.0
VIU.READ	MACHINE	routine to read voltages from a Voltage Monitor or VIU

Project 7. Barometer/Pressure Sensor (on /CHAOS2)

PRESSURE	BASIC	pressure reading program
A.TO.D.AVG.0	MACHINE	routine to read the A-to-D Converter (relocatable)
A.TO.D.AVG.SRC	ASSEMBLER	source code for A.TO.D.0
VIU.READ	MACHINE	routine to read voltages from a Voltage Monitor or VIU

Project 8. Magnetic Field Sensor (on /CHAOS2)

MAGNET	BASIC	magnetic field program
A.TO.D.AVG.0	MACHINE	routine to read the A-to-D Converter (relocatable)
A.TO.D.AVG.SRC	ASSEMBLER	source code for A.TO.D.0
VIU.READ	MACHINE	routine to read voltages from Voltage Monitor or VIU

Project 9. Smart Battery Charger (on /CHAOS2)

BATTERY	BASIC	Nickel-Cadmium Battery charge/discharge program
A.TO.D.AVG.0	MACHINE	routine to read the A-to-D converter (relocatable)
A.TO.D.AVG.SRC	ASSEMBLER	source code for A.TO.D.0
VIU.READ	MACHINE	routine to read voltages from a Voltage Monitor or VIU

Project 10. D-to-A/Digitized Sound (on /CHAOS2, /S1, and /S2)

D.TO.A	BASIC	program to control the output of the D-to-A converter
SAMPLE.PLAY	BASIC	program to use the D-to-A converter and the A-to-D converter to digitize, store, alter, and playback sounds
SP.OBJ	MACHINE	machine language routines used by SAMPLE.PLAY
WAVEFORM.MAKER	BASIC	a simple program to create waveforms using mathematical equations
miscellaneous sound files	BIN	data files of sounds ready to play using the D-to-A Converter

Project 11. Stepper Motors (on /CHAOS2)

MOTOR.TEST	BASIC	program to test stepper motor circuits (either bipolar or unipolar)
TWO.MOTORS	BASIC	program to control two stepper motors at a time (either bipolar or unipolar)

Project 12. Speed of Sound (on /CHAOS1)

SOUND	BASIC	program for timing speed of sound
SOUND.0	MACHINE	timing machine code
SOUND.SRC	TEXT	assembler source code for SOUND.0
LP.1200	MACHINE	routine to display large letters on the screen

Project 13. Ultrasonic Range Finder (on /CHAOS1)

MOTION	BASIC	start-up program. This runs program M.
M	BASIC	program to record and graph motion data
MOTION.OBJ	MACHINE	machine-language programs used by M
MDDT	BINARY	data tables used by MOTION.OBJ
LP.1200	MACHINE	routine to display large letters on the screen
INPUT	BIN	input routine
IM.HIRES.1C00	MACHINE	prints graphs of the high-resolution screen
MISC.300	MACHINE	miscellaneous machine-language routines
FONT	BIN	font used to draw letters on the high-res screen
DISTANCE	BASIC	a simple program to read and display distance
DISTANCE.OBJ	MACHINE	the routine to do the actual timing for the DISTANCE program
DISTANCE.SRC	ASSEMBLER	source code of DISTANCE.OBJ

Project 14. Chaos in the Laboratory (on /CHAOS1)

CHAOS	BASIC	Chaos timing program
FONT	BIN	font used to draw letters on the high-res screen
TIMER	MACHINE	timing machine code
IM.HIRES.1C00	MACHINE	prints graphs of the high-resolution screen

Miscellaneous

PRODOS and BASIC.SYSTEM	SYSTEM	The Apple II ProDOS operating system version 1.9. These files make a disk bootable. They are on /CHAOS1, /CHAOS2, and /S1.
STARTUP	BASIC	simple "greetings" program
UPDATE	BASIC	displays all known corrections to the book *Chaos in the Laboratory*. This file is on the /CHAOS1 disk.

Appendix E
Working with the Assembly-Language Programs

For most of the projects in this book, we have provided simple BASIC programs that you are encouraged to modify. Several projects use machine-language programs for speed. Even in some of those projects, you can often make the changes you will want in BASIC. For those of you who know 6502 machine language, or want to learn it, we have provided the assembly-language source code for several programs on the *Chaos in the Laboratory Programs* disks:

Source Code File	Object Code File	Project
PDL.CHECK.SRC	PDL.CHECK	Thermistor
A.TO.D.SRC	A.TO.D	A-to-D Converter
A.TO.D.AVG.SRC	A.TO.D.AVG	A-to-D Converter
SOUND.SRC	SOUND.OBJ	Speed of Sound
DISTANCE.SRC	DISTANCE.OBJ	Ultrasonic Range Finder

The source files were created using the Apple Editor/Assembler. They can also be used with other editors and assemblers, such as Merlin 8/16. Minor modification of the source code may be required to get the source files to assemble using other assemblers.

Using an editor/assembler package, you can load, edit, and reassemble these programs to add your changes. Always work with a back-up copy of the original disks. Make sure that you use the same name for your object file as we did; otherwise the BASIC program will not load your modified code.

Apple Editor/Assembler is part of the ProDOS Assembly Tools package sold by APDA. APDA stands for Apple Programmers and Developers Association. They distribute programming languages, books, software developer tools, and Apple system software. It is a part of Apple Computer, Inc. To order from APDA, you must have an established account and pay an annual fee.

APDA
Apple Computer, Inc.
20525 Mariani Avenue, M/S 33-G
Cupertino, CA 95014-6299
1-800-282-2732

The Merlin 8/16 Assembler comes from Roger Wagner Publishing and is sold by software stores and mail-order software suppliers.

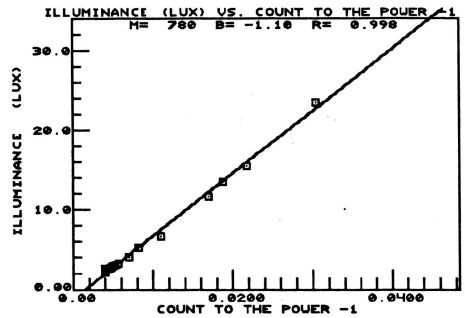
Several good books on assembly language are included in the bibliography (*Appendix C*).

Appendix F
Teacher's Guide

This section points out some of the possible problems that students may encounter with the projects. It also includes some of our test results and suggestions for ways to use the completed projects in your classroom or lab. We have included a "Feedback Form" in the back of this book. We would like to hear from students and teachers about ingenious project extensions, errors in the book, and any other comments you may have.

Introduction

The sample graph and calibration formula used in the photoresistor project were made using a Radio Shack #276-118 photocell. The calibration equation was developed by graphically determining a relationship between the PDL count and illumination. Plotting the inverse of the count vs. illumination yields this graph.



The numbers at the top of the graph are the slope, intercept, and coefficient of correlation of the linear regression "best fit" line. Note that this is a fairly linear relationship. As a result, using the equation of a line y = slope * x + y-intercept, we derive the equation illuminance = 780 * 1/count + (-1.10). Since the y-intercept was small, we ignored it for simplicity and used the equation: illuminance = 780 * 1/count.

Note that this is an approximation, and that each photocell is a little different. This equation should be fairly accurate for other Radio Shack #276-118 photocells. It will not even be close for other photoresistors. However, in many situations, you are only interested in relative light levels and hence, just the raw PDL count will be all that you need.

The fact that the resistance of a photoresistor is approximately inversely proportional to light intensity allows us to use them for Project 4 - Colorimeter.

Project 1: Thermistor Temperature Probe

The Thermistor Temperature Probe can be a useful tool for your science classes. It can be used for a number of demonstrations and experiments, including cooling curves, evaporative cooling, freezing point depression, temperature gradients in air and water and radiant heating. The nicest thing about the Thermistor Temperature Probe is that it responds very quickly to temperature change. It also has a very small thermal mass.

Another application of the Thermistor Temperature Probe is as part of a weather station built using projects from this book. For example, you could set up a two-thermistor system, built following the instructions in the "Project Extensions," and use one probe for outdoor temperature and one probe for indoor temperature. At the same time, you could use the Barometer (Project 7) or Light Sensor (Project 6). This would require merging the programs from the two chapters into one program. The final product would be a useful weather station.
Be careful when using the Thermistor Temperature Probe in chemicals. Certain chemicals will dissolve the silicone sealant or the heat-shrink tubing and ruin the probe.

The Thermistor Temperature Probe or the two-probe project extension can be used with the Vernier Software program TEMPERATURE PLOTTER III. This program includes graphing, large digit temperature display, saving and loading data, and a number of other useful features.

Project 2: Quiz Show

The finished Quiz Show project can be very useful in your classes. Kids love "playing the game." The computer monitor will indicate which team pressed one of its buttons first, making the contests more interesting as spectator events. Try using the system as part of a review session before a quiz. It also provides a great way to get something productive and entertaining done on days when half the class is missing because of weather, illness, sports, or testing.

Many schools have initiated academic quick-recall contests, either intramurally or interscholastically. In some schools these events have become major events with tryouts for the team, coaches, and playoffs. Commercial button systems that are sold for hundreds of dollars do the same job as this Quiz Show project.

You may wonder about the two resistors in the simple switch circuit. These two resistors were selected because they make the circuit work with several different computers. Minor differences in the Apple II game port digital inputs make two resistors necessary instead of just one. Even with the two resistors, a special circuit is required for the IIc, IIc+ and Laser 128 computers.

Project 3: Computer Vision

In terms of electronics and construction of the apparatus, this project is not too difficult. Construction can be greatly simplified by using toys such as Erector Sets. However, some students might want to build more complex cameras using their woodworking or metal shop skills.

The inner diameter of the tubes must be slightly larger than the diameter of the photoresistor. Be sure the students understand this before purchasing the parts, or they may have problems mounting the photoresistors. They may want to buy the photoresistors first and measure the diameter before buying the tubes.

Although this is a rather simple project, it can also be frustrating. The students might have trouble digitizing and identifying objects using the computer. Be sure that the lighting conditions are always constant. People standing around and casting shadows are often the main reason the computer is not able to identify objects. A non-glossy, black background is very helpful in increasing the contrast of an object and therefore increasing its chances of distinguishing between different objects. We have digitized the faces of twenty different people and successfully identified each of the people. A controlled environment box with an internal lighting source was used.

This project can provide students with an introduction to the topic of artificial intelligence. The Computer Vision system makes decisions about what it "thinks" the object being tested is. The rules for how it decides are programmed into the VISION program. They can be changed by students who know a little about BASIC programming. Better recognition systems can be added. The March, 1987 issue of *Byte* is an excellent introductory article on advanced image processing algorithms.

Project 4: Colorimeter

Hach Chemical Co, P.O. Box 389, Loveland CO 80539 (800) 227-4224 makes dozens of prepackaged chemicals for doing colorimetric chemical analysis. They will sell as few as 100 reagent pillows of a particular type and have no minimum order. Most of their chemicals can be stored for years without degradation. Hach also sells disposable 1 inch cuvettes (sample cells) made of plastic (part #24102-12). Twelve of these cost $18 and would be convenient if you plan to use Hach chemicals. Your colorimeter could be designed around the size of these cuvettes.

Anytime students are working around chemicals there is a hazard to both the students and the equipment. Students tend to be very causal about chemicals, so be sure to warn them of the dangers. They should always wear safety glasses, especially during the mixing operations. It is a good idea to make the colorimeter leads long enough so the device can be several feet from the computer. A chemical spill into the keyboard would be a disaster. In most schools there are one or two computers per classroom. Under these conditions it is a good idea to have the students prepare the chemicals away from the computer and bring them to the computer only when they are ready to read the absorbance.

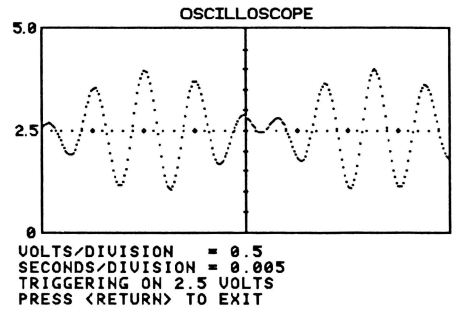
The dangers of power tools cannot be overemphasized. The shop teacher, a carpenter, or a parent who happens to have power tools should be willing to help.

Photoresistors are inexpensive, easy-to-use light sensors. They have some limitations that should be kept in mind as you use your colorimeter. The response of a photoresistor depends on temperature. This is not a major effect, but it could cause problems if your lab temperature changes very much. A typical photoresistor's data sheet reports that a 20°C change in temperature causes an 8% change in resistance. Another problem is the light history effect of photoresistors. A photoresistor that is exposed to bright light for a while will respond differently than one that is kept in the dark. Neither of these effects should be too important using this colorimeter, as long as you do all experiments within a relatively short time interval and use LEDs as your light source.

Some teachers may wonder why we use the 7404 inverter IC. The IC adds to the complexity of the project, and does not seem to really do anything. Sometimes the annunciator output lines on the Apple II game port are used to control LEDs directly. This can be done in some cases, especially with the older Apple II models. However, powering LEDs directly off the annunciator lines involves more current than these lines are designed to handle. We use the 7404 as a buffer to sink the current required for operating the LEDs. In this way, we are not exceeding published limits set by Apple Computer, Inc. for the annunciator lines.

Project 5: A-to-D Converter

This project provides an inexpensive way of monitoring voltages with your computer. It is a key project, since it can be used in four other projects in this book. The SCOPE program can be used for a number of interesting demonstrations with sounds. If you connect the microphone described in the "Project Extensions" section to the A-to-D Converter, students can compare the "shapes" of their sound waves. Connecting an electronic keyboard (such as a Casio SK-1) as the input device allows you to demonstrate the sound wave patterns made by several different instruments. The flute sound made by electronic keyboards is usually a very pure sine wave. Playing two notes at the same time will demonstrate beats nicely.



A Sample Beat Pattern

The SCOPE program reads only the seven highest bits of the voltage, even though 10 bits are available. There are two reasons for this: (1) The voltage is displayed as a dot on the Apple II screen and there are only 191 vertical pixels on the screen. Reading the voltage to more accuracy would not be productive, since the display cannot have more than 191 possible values. (2) Reading only the first seven bits speeds up the program a little.

We strongly recommend building or purchasing the microphone described in the project extensions. Not only is it useful for showing sound wave patterns using this project, but it can allow your students to digitize sounds using Project 10. All the parts for this microphone are available at Radio Shack and both assembled microphones and microphone parts kits are available from Vernier Software.

In the electrochemical potential project suggested in the "Step-by-Step Instructions," the voltages you can expect will vary with temperature and other conditions but they should be similar to the following:

Lemon with copper wire and a nail.....0.48 V with the copper as the positive
Nickel/penny sandwich (vinegar) .....0.28 V with the copper as the positive
Aluminum/copper (baking soda) .....0.98 V with the copper as the positive
Zinc/copper (salt).....0.80 V with the copper as the positive

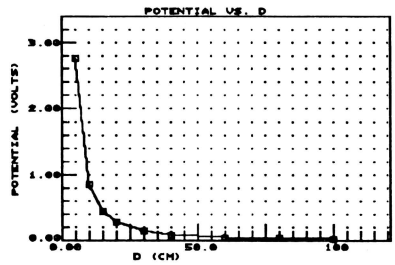
In the wire-smashing experiment, students need safety glasses and a few words about moderation. If the wires are hit too hard they will break and no data will be collected. Very hard wire such as nickel or chrome will not work as well as softer metals such as copper or steel. The signal is small, so a preamplifier will probably be necessary. The program SCOPE can be used to measure and display fast signals.

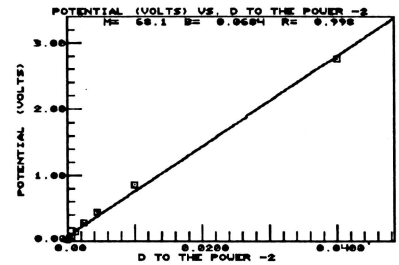
Project 6: Light Meter

The Light Sensor should be useful for many experiments and demonstrations. Here are some sample results.

Inverse Square Law Experiment

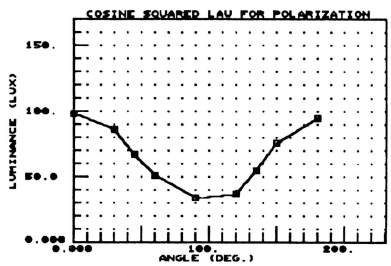
A sample inverse square law experiment results is shown in the graphs below (graphed using Vernier Software's GRAPHICAL ANALYSIS III program). The graph on the left is voltage out of the Light Sensor vs. distance from the source. The source was a small lamp. The graph on the right shows the output voltage plotted vs. $1/d^2$. Note that this graph shows that the output seems to drop off in accordance with the inverse square law.

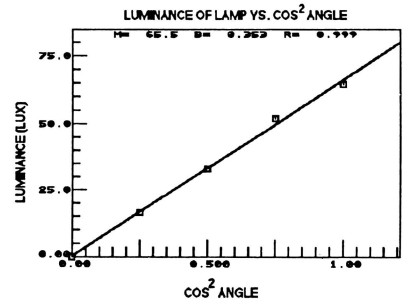




Malus' Law for Polarizing Filters

The Law of Malus is explained in most college physics texts. It states that the intensity of light transmitted through two polarizing filters varies according to this equation: $I = I_m \cos^2 \emptyset$, where $\emptyset$ is the angle between the polarizing directions of the two sheets. The two graphs below show experimental data taken with the Light Meter. The graph on the right shows that a plot of illuminance vs. $\cos^2 \emptyset$ is nearly a straight line.





Comparison between Vernier Light Meter and INS Light Meter

To test the accuracy of this Light Meter, we did a comparison between it and a commercial light meter. The Light Sensor and an INS Light Meter (which is available through Edmund Scientific) were held next to each other in various lighting situations. The results are shown below.

Situation	Vernier Meter (lux)	INS Meter (lux)
Palm of hand	1	1
2 cm from a faint light	384	343
20 cm from a fluorescent lamp	5470	4400
2 cm from a computer screen	66	70
On top of desktop in well-lit room	743	591
Background light in room	156	122
Under a table (darkened area)	29	20

Note that this project uses a 5-pin DIN plug, and can be plugged into and used with a number of other laboratory interfaces:

- Vernier Software Voltage Input Unit (Apple II or IBM-compatible) and the VOLTAGE PLOTTER program
- Vernier Software MultiPurpose Lab Interface (Apple II or IBM-compatible) and the MPLI Program
- Universal Lab Interface (Macintosh) and the Data Logger program

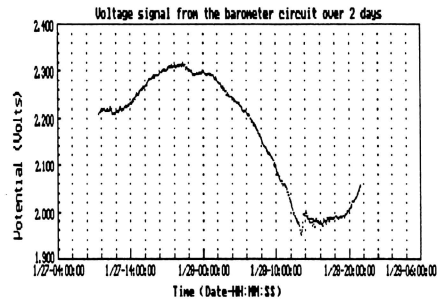
These lab interfaces all provide for calibration, data collection, graphing, data storage and retrieval.

A Vernier Software 16-pin Voltage Input Unit can be used with the LIGHT program with no changes. Choose "V" when asked for the type of voltage input device being used. If you have a 9-pin Voltage Input Unit, you need to make one minor program change before you can use the VIU with the LIGHT program. Change line 165 to read: `165 CH=0: REM SET FOR PB0 INPUT.`

An assembled version of the Light Sensor circuit in a box, with the sensor mounted at the end of a plastic probe, is sold by Vernier Software for $39.00 (Order Code LS-DIN).

Project 7: Barometer / Pressure Sensor

This project can be the start of a computerized weather station. Air pressure is probably the one weather parameter that is the best predictor of future whether trends.[2] This graph shows the pressure changes over several days. This graph was made using the Vernier Software program VOLTAGE PLOTTER III. It shows a weak low-pressure area moving in.

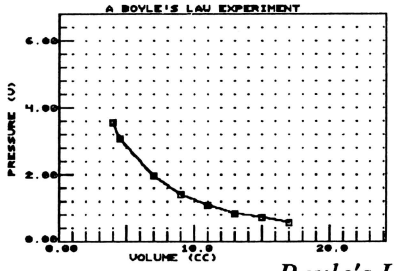


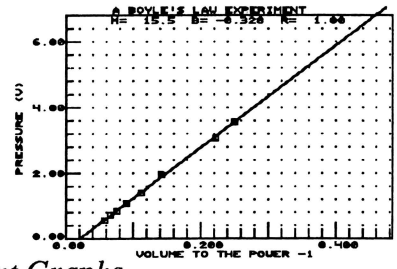
The pressure sensor used here is designed for use in air and other non-corrosive gasses only. Be sure to warn students to avoid getting water inside it during experiments with water-filled tubes. If the sensor does get wet, try dumping all of the water out of it and letting it air dry.

There are many other experiments and demonstrations that you might want to try with this project. Boyle's law experiments using large plastic hypodermic syringes is a good example. Use a short

[2]This sentence includes the worst spell of weather we have had in a long while.

piece of tubing to connect the syringe to the sensor. Graphs of some sample data are shown below. The graph on the left is the raw data and the one on the right shows the inverse relationship.





Boyle's Law Experiment Graphs

Other experiments and demonstrations that were not mentioned in the chapter include:

- Demonstrating Bernoulli's principle
- If you have a multi-story building available, you might try to demonstrate the change in pressure with altitude. The change is roughly 0.01 inch of Hg per 100 feet, so you have to change the altitude quite a bit to see the effect.

Teachers are encouraged to contact SenSym to obtain their *Pressure Sensor Handbook.* This is their catalog, but it contains a lot of useful information on pressure measurement.

Here are some questions related to pressure that you may want to discuss with your students:
- Does the pressure measured inside a building accurately represent the pressure outside? How can you tell?
- Can you make a calculation to predict wind speeds based on pressure differences?
- If two air masses are alike in all respects except their relative humidity, the air mass with the higher humidity will have the lower pressure. Why?

Note that this project uses a 5-pin DIN plug, and can be plugged into and used with a number of other laboratory interfaces:

- Vernier Software Voltage Input Unit (Apple II or IBM-compatible) and the VOLTAGE PLOTTER program
- Vernier Software MultiPurpose Lab Interface (Apple II or IBM-compatible) and the MPLI Program
- Universal Lab Interface (Macintosh) and the Data Logger program

These lab interfaces all provide for calibration, data collection, graphing, data storage, and retrieval. In each case, you would need to do a two-point calibration so that the program reads data in pressure units instead of volts.

A Vernier Software 16-pin Voltage Input Unit can be used with the PRESSURE program with no changes. Choose "V" when asked for the type of voltage input device being used. If you have a 9-pin Voltage Input Unit, you need to make one minor program change before you can use the VIU with the PRESSURE program. Change line 165 to read: `165 CH=0: REM SET FOR PB0 INPUT`

An assembled version of the Barometer/Pressure Sensor circuit in a box is sold by Vernier Software (Order Code BAR-DIN).

Project 8: Magnetic Field

The Magnetic Field Sensor has lots of possibilities for labs and demonstrations. Several of these are listed in the project extensions. Students may find it interesting to measure magnetic fields directly with an instrument. In most cases they will never have done this before.

There are several traditional physics labs that involve measuring magnetic field intensity indirectly.

Two examples are:

- Measuring the field around a current-carrying wire using small compasses (*PSSC Physics* "The Magnetic Field Near a Long, Straight Wire")
- Measuring a magnetic field by determining the force it produces on a sensitive current-carrying balance (*PSSC Physics* "The Measurement of a Magnetic Field in Fundamental Units" or *Project Physics* "Forces on Currents.")

As a follow-up to either of these experiments, it would be interesting to compare results with the Magnetic Field Sensor readings for the same field.

The Magnetic Field Sensor can be used to show students the earth's magnetic field. With the core project the earth's field is just barely measurable. If you do the project extension of increasing the amplifier gain, the earth's field can be accurately measured. You can determine the direction of the field by looking for a maximum in the magnetic flux reading. The readings will be largest when the green side of the transducer points toward the north magnetic pole. Many students may be surprised that the magnetic inclination is so large.

Exploring the field near a color monitor can be a good demonstration. Use the A-to-D Converter and the SCOPE program. In some cases you can see a sawtooth wave form, which is caused by the electromagnets used to move the electron beam up and down on the monitor screen.

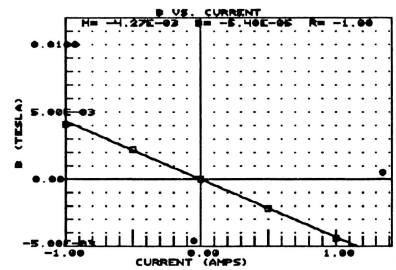
Measuring the field inside a solenoid or loop and comparing it with the theoretical value given by textbook equations makes another good demonstration or lab. Here are some sample results using a solenoid with a radius of 3.0 cm, 15.2 cm long, with 570 turns:

Theory: $B = \mu_0 I n = (1.26 \times 10^{-6})(570 / 0.152 \text{ m})(i) = 4.7 \times 10^{-3} \text{ tesla} * i$

where: B = flux density in tesla
μ = permeability constant = 1.26×10^{-6} H/m
i = current
n = number of turns of wire

Experimental:

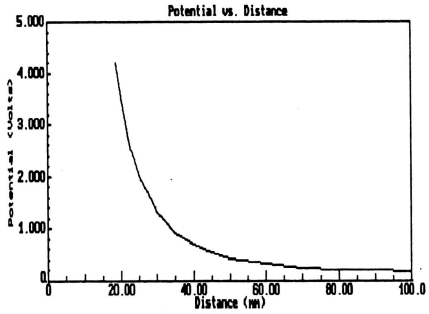
current (amps)	B (tesla)
0	0
-0.5	2.21×10^{-3}
-1.0	4.16×10^{-3}
+0.5	-2.24×10^{-3}
+1.0	-4.38×10^{-3}

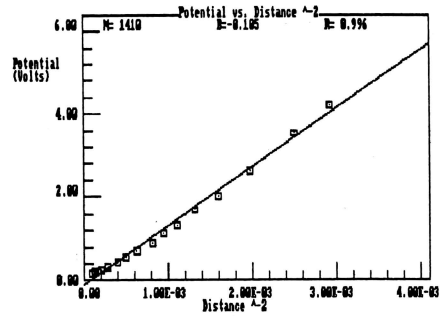


Magnetic Field vs. Current For a Solenoid

The data is graphed at the right. The slope of the line shows that B changed by 4.3×10^{-3} T for each 1 amp change in current. The change in field with current was very linear. The field is about 9% smaller than predicted by the ideal solenoid equation. Some of this difference may be due to the fact that the solenoid is fairly short relative to its radius; that is, it was not a very good approximation of an ideal solenoid.

The graphs below show the results of investigating how a magnetic field drops off as you move away from a permanent magnet as described in the "Project Extensions." The graph on the left is the raw data (B vs. distance); the graph on the right shows B vs. 1/distance squared.





Inverse Square Law Experiment Graphs

Two good sources of information on Hall effect sensors are:

Hall Effect Transducers, How to Apply Them as Sensors, Micro Switch, A Honeywell Division, 1982, Freeport, IL 61032

Hall Effect and Optoelectronic Sensors Data Book SN-500, Sprague Electronics, Concord, NH.

Both of these include background information on the sensors and application notes.

Note that this project uses a 5-pin DIN plug, and can be plugged into and used with a number of other laboratory interfaces:

- Vernier Software Voltage Input Unit (Apple II or IBM-compatible) and the VOLTAGE PLOTTER program
- Vernier Software MultiPurpose Lab Interface (Apple II or IBM-compatible) and the MPLI Program
- Universal Lab Interface (Macintosh) and the Data Logger program

These lab interfaces all provide for calibration, data collection, graphing, data storage, and retrieval. In each case, you would need to do a two-point calibration so that the program reads data in units of tesla or gauss instead of volts.

A Vernier Software 16-pin Voltage Input Unit can be used with the MAGNET program with no changes. Choose "V" when asked for the type of voltage input device being used. If you have a 9-pin Voltage Input Unit, you need to make one minor program change before you can use the VIU with the MAGNET program. Change line 165 to read: `165 CH=0: REM SET FOR PB0 INPUT`

An assembled version of the Magnetic Field Sensor circuit in a box, with the sensor mounted at the end of a plastic probe, is sold by Vernier Software for $44.00 (Order Code MG-DIN).

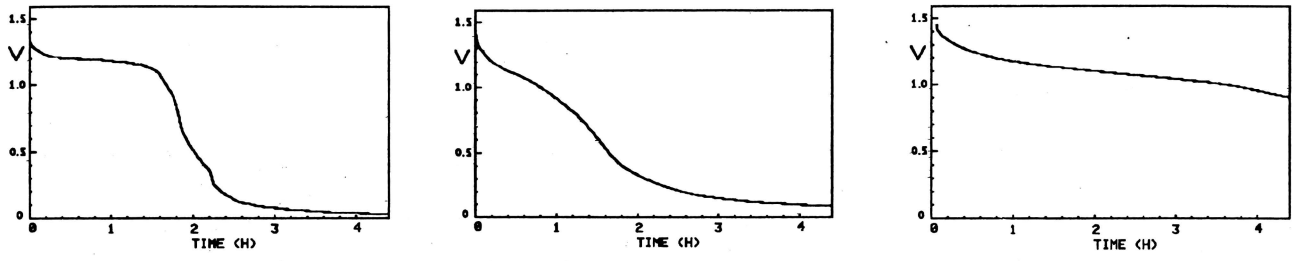
Project 9: Smart Battery Charger

This project can be very worthwhile if you are a nickel-cadmium battery user. We find it useful in maintaining the many batteries used in miscellaneous hand-held instruments, electronic keyboards, and flashlights. Some of your students may be involved in racing radio-controlled model cars. These students may have a special interest in this project. One of the most important factors in this kind of racing is to maintain your batteries in the best of condition. One bad battery in a set used for a race could cost the operator a victory. There are services available to model car racers that will test batteries, charge them, and make them into matched sets for best performance. All of this can be done using Smart Battery Charger.

You may want to get in the habit of leaving a battery in the Smart Battery Charger when you go home at night. In the morning, you will have a charged battery and a graph reporting the status of the battery.

Since the recharging of nickel-cadmium batteries takes a long time, Smart Battery Charger is designed to allow you to continue to charge batteries while you use your computer for other things. You can even turn the computer off. If you disconnect the Smart Battery Charger circuit from the game port during use with the external power supply still on and the battery in place, the circuit will be in the charge mode. This means that if you need to use the computer while Smart Battery Charger is being used to charge a battery, you can disconnect the circuit from the computer and the battery will continue to charge. Of course, your graphs of the charging will be lost. Turning off the computer has the same effect. Remember that there is no harm done when a battery is left in the circuit with a slow charge being applied for extended time periods.

Many students may be interested in doing a consumer study of batteries. Which types and brands are the best value? The Load Test can be used for this purpose. Remember that certain batteries are best in specific uses—heavy loads, light loads, long-term storage, etc. If you compare just nickel-cadmium batteries, you can use the circuit as is. You can also test disposable batteries with the Smart Battery Charger discharge circuit and the Load Test option. It is probably a good idea to disconnect the charging resistor if you use the Smart Battery Charger with disposable batteries, to avoid the chance of accidentally recharging them. Attempting to recharge batteries that were not designed for that purpose can cause explosions.



Discharge Graphs for Nickel-Cadmium, Carbon, and Alkaline Batteries

This is probably the most difficult project in this book. The "Step-by-Step Instructions" are not as complete as in most of the other projects. The students have to make more decisions about how to position the components on the circuit board. Also, they need to understand Ohm's law and calculate the proper resistors to use. Only students with some experience in electronics should be encouraged to work on this project.

This project can be done using a 16-pin Voltage Input Unit (VIU) as the voltage input device. The 16-pin VIU is equivalent to a Voltage Monitor, and the BATTERY program can be used as shipped. If you have a VIU with a 9-pin connector on it, you need to make a one-line change in the BATTERY program. Change line 165 so that it reads: `165 CH=0: REM CHANGE THE INPUT LINE.` Always make your changes on a back-up copy of the disk, and keep the original in a safe place.

Project 10: D-to-A / Digitized Sound

We have had lots of fun with this project, and we would bet you and your students will also. By all means be sure to set up a speaker and listen to the sounds. Considering the simplicity of the hardware, it is surprising how good the sound is. Digitizing your own sounds is easy if you build the microphone described in Project 5. Both parts kits and assembled versions of this microphone are available from Vernier Software.

The HIDDEN.MESSAGE sound on the /S2 disk is a sound digitized backwards. It is not satanic. It can be played backwards by the SAMPLE.PLAY program and it will sound normal.

The Digital-to-Analog Converter may serve a useful purpose as a laboratory function generator. If you are willing to modify the WAVEFORM.MAKER program, you can program any wave shape you can mathematically generate.

Your students will go through batteries fairly quickly when using this project. You may want to use a power supply or rechargeable batteries to save expense and minimize waste.

Project 11: Stepper Motors

This is not an easy project, and is therefore not suggested as a first project for a student. Here are some suggestions that may help ensure success:

- Unipolar stepper motors with the UCN-5804B driver chip seem to be the best choice. The IC is more difficult to damage, and only a 5-volt external power supply is required. Also, the stepper motors available from Vernier Software have identified leads, so the assembly will be considerably easier than with a motor with unknown leads.
- If you are getting your own unipolar stepper motor, be sure it has a current rating below 1.5 amps. Determining the leads will be a trial-and-error process, and there are a lot of combinations.

  If you are using a bipolar stepper motor, be sure it has a current rating below 350 mA. Exceeding these limits could damage the stepper motor controller chip. Identifying the leads on a bipolar stepper motor is not difficult. Again it will be a trial-and-error process, but this time there are only two possibilities after you do some tests with an ohmmeter.
- If you plan to use two stepper motors, it is a good idea to get one motor working correctly before connecting the second.

Projects involving multiple stepper motors will require a substantial power supply. Be sure your external power supply has a current output rating that is large enough for your stepper motors. Stepper motors have a holding current that is nearly as great as the operating (running) current. Most inexpensive power supplies can supply about 1 to 2 amps, which is only enough current for three stepper motors (maybe less, depending on the rating of the motors).

Stepper motors are designed to give precise rotation, but they are not very powerful. The hardware associated with this project must be designed to operate without binding or excessive friction. If friction causes the motor to miss a step, the precision associated with stepper motors is lost.

Stepper motors are available with a variety of steps per rotation. Be sure to order motors with lots of steps per rotation for precision work, and those with few steps per rotation if you need faster motion. Many stepper motors are sold with gear reduction systems on them to provide many more steps per rotation.

The bipolar stepper motor circuit described in this project is designed around 12-volt stepper motors. We have used the same circuit with a 5-volt external power supply and 5-volt bipolar stepper motors. The reason we do not promote the use of 5-volt bipolar stepper motors is that the MC3479P data sheets say it requires at least 7.2 volts to operate. Using our bipolar stepper motor controller circuit with a 5-volt external supply will mean powering the MC3479P with just 5 volts. It does seem to work in our tests.

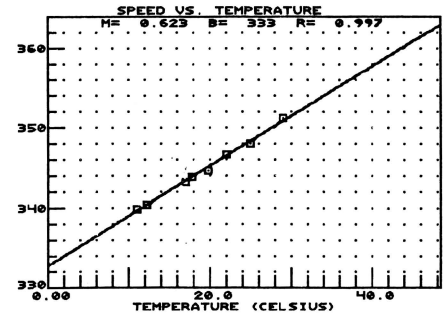
Project 12: Speed of Sound

This project can make a good, quick demonstration of the speed of sound in your science classes. Use the large digit display so that students can see the times as they are measured. Have them measure the distance and calculate the speed. Move the second microphone to another position and show them that the time changes, but that the speed calculated comes out the same. Students may bring up the fact that the electrical signal has to travel a long distance in the wires. This should lead to a discussion of the difference in the speeds of sound and electrical signals.

There are many other ways to measure the speed of sound, but most of them are indirect, relying on resonance in a tube or looking at traces on an oscilloscope. For many students, this experiment may be much easier to understand since you are really measuring the time it takes sound to travel from one

place to another. Some students may be impressed that the computer can measure such a short time. Actually, by computer standards, these times are fairly long.

As an example of a extended experiment you might want to conduct using this project, this graph shows measurements of the speed of sound made outdoors. Timings were made during the day, as the temperature increased, over a range of 5 meters. The change in the speed of sound for 1°C change in temperature is often given as 0.59 m/s.[3] The graph shows a linear fit on data we gathered using the Speed of Sound circuit. It gives a slope (M) of 0.623 m/s°C, which is fairly close to the accepted value. The y-intercept on the graph of the speed of sound in air at 0°C came out to be 332 m/sec, which is fairly close to the accepted value for the speed of sound at 0°C.



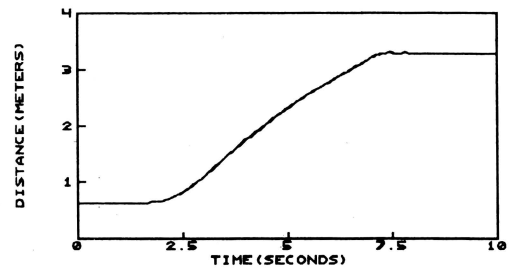
Speed of Sound in Air vs. Temperature

Some of our test sites for this project have reported getting very good results in measuring the speed of sound in aluminum rods. The microphones were attached to the side of the rod. The rod was placed on a soft surface and a small mallet was used to tap the end. Speeds of about 4900 m/s were measured.

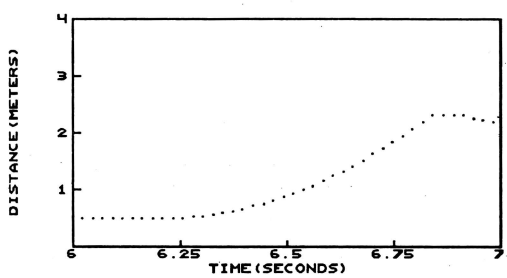
If you have trouble with this project, be sure to read the "Troubleshooting" section. It explains how this project can initiate the Apple IIe or IIGS self test under certain conditions.

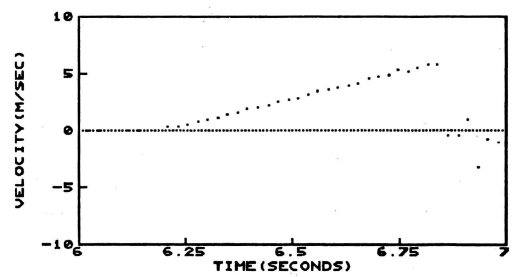
Project 13: Ultrasonic Range Finder

The final product of this project can be very useful in your science classes. Students usually love to use themselves as "objects" and study the graphs made while they move toward or away from the Range Finder. Numerous studies have shown that this kind of experimentation is a very good way to learn physics.[4] This program will not do real-time velocity graphs, but you can collect data in real time as you graph distance vs. time and then plot the velocity vs. time graph for comparison. This can be a good tool in teaching students about kinematics graphs. This graph shows an attempt to produce constant velocity for a few seconds.



The Range Finder can even be used to study free fall. The graphs below show the motion of an object falling about 2 meters. Note that the choice of minimum and maximum values on the axes have been used to expand the graph so that only the time of the fall is graphed.





Distance vs. Time and Velocity vs. Time Graphs for Free Fall

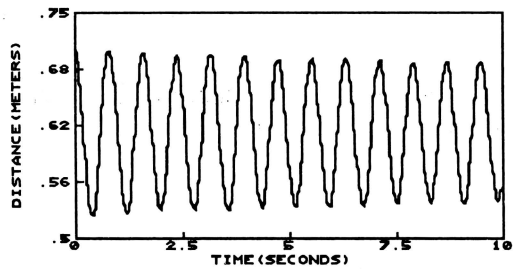
---

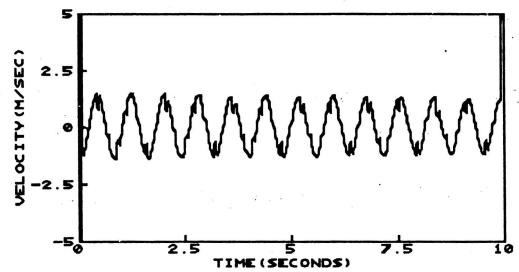
[3]*Handbook of Chemistry and Physics*, 45th Edition, 1964, The Chemical Rubber Company, Cleveland, OH.
[4]See, for example, the article by Thornton and Sokoloff cited in the reference section of this project.

Range finders of this type have been used in physics since about 1984. Many of the early ones were developed at the Technical Education Research Centers (TERC). Workshops involving construction of Range Finders have been run by several groups. Now, commercial Range Finders are available for most of the popular personal computers.[5] They represent one of the best ways to illustrate the connection between a graphical representation of motion and the motion itself. A moving person can see their position vs. time graph as the motion takes place. This is a very powerful tool!

The useful range of the device is 40 cm (1.33 feet) to about 10.7 meters (35 feet). The limits and accuracy of the Range Finder are affected by the acoustics of the area in which it operates. If an object is too close, the ranger will not measure the distance because the return will come during the blank echo period. If it is too far, the return signal will be too weak to receive. It can also pick up the position of the wrong object if the area contains other objects in the path of the signal, so be careful to operate the Range Finder in a place where it will pick up only what you wish. The Range Finder also does not work well close to operating color televisions or any other electronic devices that produce 50-kHz ultrasound. Recognizing these characteristics is essential for success with the Range Finder.

The Range Finder makes a great way to study simple harmonic motion. The sample graphs below were made using a 1-kg mass suspended from a spring and hung directly above the Range Finder.





Graphs of Simple Harmonic Motion Made with the Range Finder

Students should be warned not to touch the inside of the transducer or the circuit board while the Range Finder is operating. A high-voltage is applied to the transducer, and if it is touched by a person, he or she will get a shock. This will not do any real damage, but it is slightly painful. It is similar to the shock you get when you get a strong static discharge after walking across the rug on a dry, cold day. The plastic housing of the Range Finder prevents any problems unless the students open the unit while it is operating.

Project 14: Chaos in the Laboratory

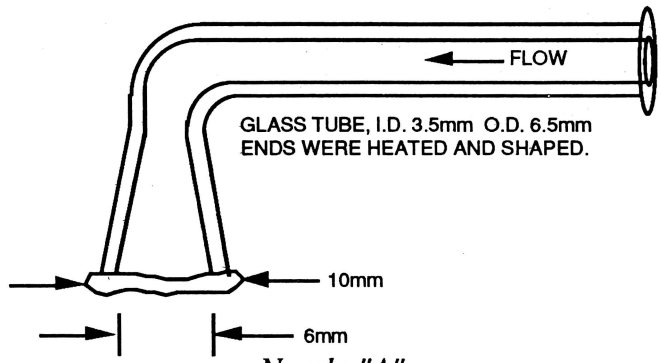
As you probably know, chaos has been a popular topic in all areas of science and math in recent years. It has been the topic of many magazine and newspaper articles, a best-selling book, and a NOVA program. Much serious research is being done in the subject. There is even a new journal published by the American Institute of Physics called *Chaos*, which is devoted totally to the subject. This project allows an interested student to do a little research in the field.

This is a nice project for students who tend to be experimentally oriented and have the potential to be inspired to learn abstract concepts. Obtaining clear and recognizable return plots of strange attractors can be a motivating experimental challenge that may be expanded to include different nozzles, fluids, temperatures, etc. Underlying these intriguing images is the theoretical challenge of finding a simple equation that when plotted on the classroom computer will yield similar pictures. It has been my experience that many students who learn best with their "hands on" need a motivating application when it comes to exploring the abstract. It is in that spirit that this project has been designed.

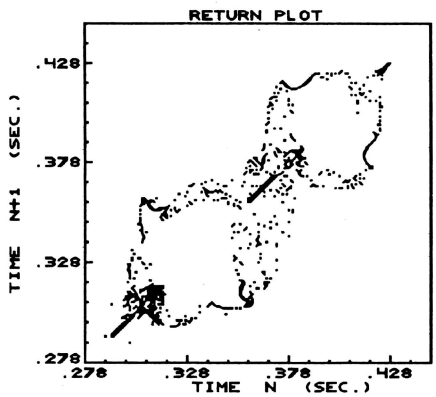
[5] A number of different suppliers are mentioned in the article by Risley cited in the reference section of this project.

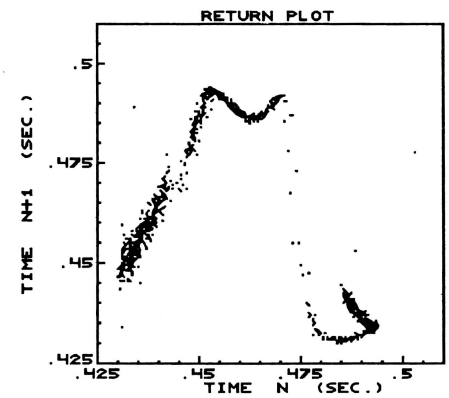
Much can be learned about the science of a dripping faucet by thinking of what kind of mathematical model could describe the motion of the drip as it forms, and then predict when it will drip. Simple equations can yield data that may produce images similar to those obtained by this project. By carefully watching the drops form and let go, one might notice how they wiggle and how the vibration left behind by a drop can affect when the next one lets go. While repeating period one and period two patterns may be due to a predominating relationship between the fill rate and the vibrations of the filling drop, it has been shown by Shaw (see "Bibliography") that the chaotic structure is due to a communication between drops. What is it that preserves part of what was happening in one drop and passes it on to the next? It is possible to get into a very complex discussion about that relationship, but that is beyond the scope of this project.

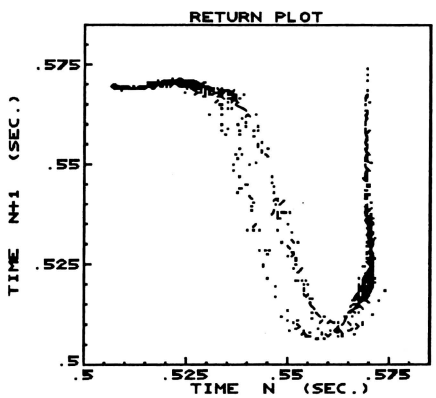
Different nozzle shapes can have a marked effect on the results. The map below and the following strange attractors were made using nozzle "A". The map is a more detailed look at a range of drip times, obtained by using a slower rate of change of flow. The relative flow rate of each plot is related to the mean time between drops. The physical description of nozzle "A" appears below:

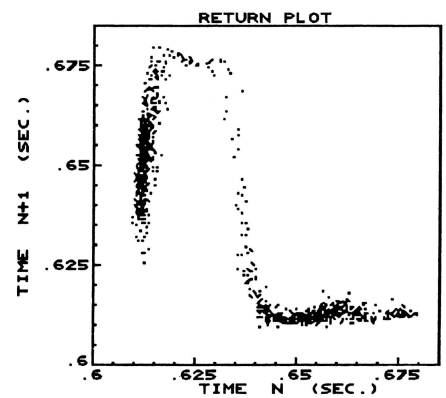


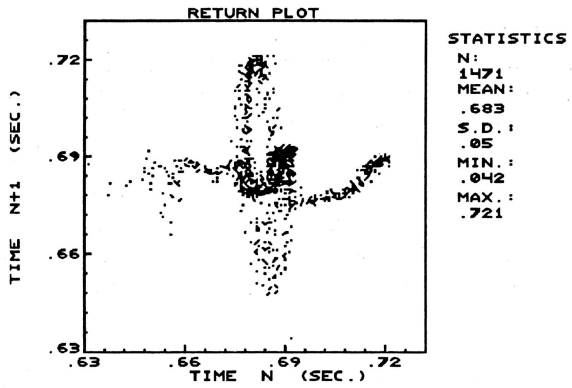
Nozzle "A"

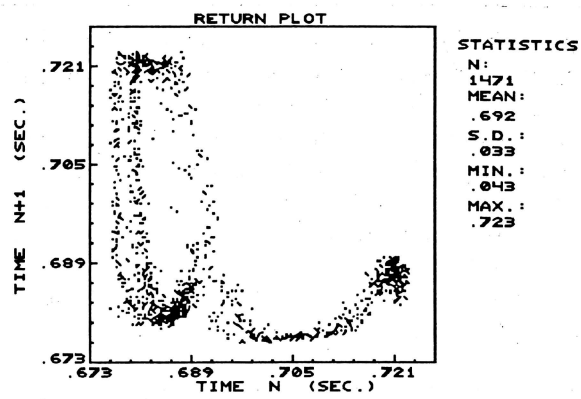












Return Plots Made with Nozzle A

Recommended reading for all students is James Gleick's book *Chaos: Making a New Science*, and his article in *Smithsonian* magazine. The excellent series of articles found in *New Scientist* is a basic and comprehensive review of chaos theory and its many applications (see "Bibliography"). I also highly recommend viewing the NOVA program *The Strange New Science of Chaos,* which highlights the dripping faucet and other applications of chaos theory.

Appendix G
Other Vernier Software Products

Vernier Software has been publishing science software since 1981. We have software and hardware for Apple II, IBM-compatible, and Macintosh computers. The following is a list of products available. If you would like to receive a complete catalog of Vernier Software products, write or call:

Vernier Software
2920 S.W. 89th Street
Portland, OR 97225
(503) 297-5317 FAX (503) 297-1760

Books on Computer Interfacing

◊ ***How to Build a Better Mousetrap and 13 Other Science Projects Using the Apple® II:*** This 227-page book includes 14 science projects for students (or teachers) who want to learn about laboratory interfacing via the Apple II+/IIe/IIGS game port. Projects include: temperature probe, pH meter, strain gage, stepper motor, and photogate timer. $24.95

◊ ***Interfacing the High School Physics Lab*** by John N Fox. A collection of experiment and demonstration ideas using our MPLI hardware and software. The ideas come from a series of summer workshops for high school teachers. Physics, chemistry and biology topics are included.

◊ ***Chemistry with Computers*** by Dan D. Holmquist and Donald L. Volz. A collection of experiments and demonstration ideas using Vernier Software programs and probes. Either the Apple II or MS-DOS versions of our products can be used. The student labs sheets are provided on AppleWorks and MS-DOS disks so that teachers can edit and print them.

Software and Hardware for the Apple II

◊ **MULTIPURPOSE LAB INTERFACE PROGRAM:** Turns your computer into a three channel data recorder and storage oscilloscope. Package includes program, analog-to-digital board, and connector box. $290.00

◊ **GS SCOPE:** Turns your Apple IIGS into a three channel storage oscilloscope. This program takes full advantage of the IIGS speed, graphics, and mouse. Requires our Apple II MPLI package. $39.95

◊ **TEMPERATURE PLOTTER III**: Provides an accurate, inexpensive method of monitoring temperatures with your Apple II. Requires a temperature probe system ($25 to $70) which connects to the game port. $39.95

◊ **VOLTAGE PLOTTER III**: Allows the Apple II to be used as a general purpose voltmeter or chart recorder. Requires a Voltage Input Unit ($45) which connects to the game port. $39.95

◊ **GRAPHICAL ANALYSIS III**: Plots graphs of experimental data and allows students to examine the data in a number of ways. $29.95

◊ **PRECISION TIMER III**: Turns the Apple II into a laboratory timer. Requires a photogate system ($40 for a 3-Photogate Parts Kit) which connects to the game port. $39.95

◊ **FREQUENCY METER III**: Allows the Apple II to be used as a general purpose audio-range frequency meter in the laboratory. Requires a microphone/amplifier ($30) which connects to the game port. $39.95

◊ **MILLIKAN OIL DROP EXPERIMENT:** A simulation of one of the classic physics experiments. $29.95

◊ **VECTOR ADDITION III**: Graphically demonstrates the head-to-tail addition of vectors. $29.95

◊ **RAY TRACER**: Illustrates the principles of geometrical optics. $24.95

◊ **ORBIT II**: Simulates the motion of an earth satellite. $24.95

◊ **CHARGED PARTICLES II**: Simulates the motion of various charged particles in magnetic and electric fields. $24.95

◊ **PROJECTILES II**: Allows students to experiment with projectile motion, including the effects of air resistance and wind. $24.95

◊ **WAVE ADDITION III**: Graphically demonstrates the superposition of waves. $29.95

◊ **KINEMATICS II**: Practice kinematics problems for beginning physics students. $24.95

Software and Hardware for IBM-Compatible Computers

◊ **MULTIPURPOSE LAB INTERFACE PROGRAM (MS-DOS® Version):** Turns your computer into a three channel data recorder and storage oscilloscope. The IBM MPLI hardware-software package consists of a 12-bit, 3-input A/D board, an interface box, and program. $310.00

◊ **TEMPERATURE PLOTTER (MS-DOS® Version)**: Provides an accurate, inexpensive method of monitoring temperatures with your IBM. Requires a temperature probe system which connects to an IBM game port. $39.95

◊ **VOLTAGE PLOTTER (MS-DOS® Version)**: Allows the IBM to be used as a general purpose voltmeter or chart recorder. Requires a Voltage Input Unit ($45) which connects to an IBM game port. $39.95

◊ **PRECISION TIMER (MS-DOS® Version)**: Turns the IBM PC or the IBM-compatible computer into a laboratory timer. Requires a photogate system ($60 for a 4-Photogate Parts Kit) which connects to an IBM game port. $39.95

◊ **FREQUENCY METER (MS-DOS® Version):** Allows the IBM to be used as a general purpose audio-range frequency meter. A microphone/amplifier ($30.00), which connects to an IBM game port, is used to deliver the signal to the game port. $39.95

◊ **GRAPHICAL ANALYSIS (MS-DOS® Version)**: Plots graphs of experimental data and allows students to examine the data in a number of ways. $29.95

◊ **MILLIKAN OIL DROP EXPERIMENT (MS-DOS® Version):** A simulation of one of the classic physics experiments. $29.95

◊ **ORBIT (MS-DOS® Version)**: Simulates the motion of an earth satellite. $29.95

◊ **CHARGED PARTICLES (MS-DOS® Version)**: Simulates the motion of various charged particles in magnetic and electric fields. $29.95

◊ **PROJECTILES (MS-DOS® Version)**: Allows students to experiment with projectile motion, including the effects of air resistance and wind. $29.95

Universal Lab Interface for the Macintosh

The Universal Lab Interface (ULI) connects to any Macintosh computer via the modem port. It allows a wide variety of sensors and probes to be interfaced to the Macintosh. Programs are available to monitor, graph and save distance, velocity, acceleration, force, temperature, voltage, pH, count rates, and timing data.

Parts Kit Order Form

Vernier Software produces parts kits for several of the projects in this book. You may find these parts kits convenient because some of the components for these projects are difficult to obtain in single quantities. Each kit contains all of the necessary components (listed in the parts list at the beginning of the project). Parts kits that connect directly to the computer will include a game port cable. In some instances you have a choice of either a 9-pin cable or a 16-pin cable. The price for each kit is listed below.[6] Add 3% shipping per order ($4.00 minimum). Please enclose either school purchase order or check.

Project (Order Code)	Parts Kit Price	Quantity	Amount
Thermistor Temperature Probe			
with 9-pin cable (CTP-9)	$10.00	_____	_____
with 16-pin cable (CTP-16)	$10.00	_____	_____
A-to-D Converter (CAD-16)	$32.00	_____	_____
Light Meter (CLM-DIN)	$16.00	_____	_____
Barometer / Sensor (CBR-DIN)	$42.00	_____	_____
Magnetic Field Sensor (CMG-DIN)	$24.00	_____	_____
D-to-A / Digitized Sound (CDA-16)	$21.00	_____	_____
Stepper Motors - Unipolar (CSU-16)	$16.00	_____	_____
Speed of Sound			
with 9-pin cable (CSS-9)	$19.00	_____	_____
with 16-pin cable (CSS-16)	$19.00	_____	_____
Chaos in the Laboratory			
with 9-pin cable (CCH-9)	$12.00	_____	_____
with 16-pin cable (CCH-16)	$12.00	_____	_____
Other Items (Order Code)			
Chaos in the Laboratory...(CHAOS)	$25.95	_____	_____
How to Build a Better Mousetrap...(MT)	$24.95	_____	_____
Microphone for use with A-to-D Converter			
parts kit (MIC-U)	$15.00	_____	_____
assembled (MCA-U)	$30.00	_____	_____
Game Port Extender (GPE)	$16.00	_____	_____
Shipping 3% ($4.00 minimum)			_____
		Total-->	_____

☐ Please send me a catalog of Vernier Software products.

Ship to:

Name_____

School_____

Address_____

City, State, Zip _____

Send order to:

Vernier Software
2920 S.W. 89th Street
Portland, OR 97225
(503) 297-5317
FAX (503) 297-1760

[6]Prices and availability of these parts kits are subject to change.

Feedback Form

We would like to hear from you. Please contact us if you have comments or suggestions about this book.

Suggestions for improving the book (corrections, clarifications, etc.):

Interesting project extensions you have developed:

Other comments:

Name _____

School _____

Address _____

City, State, Zip _____

I am a: ☐ Student ☐ Teacher

Send to:

Vernier Software
2920 S.W. 89th St.
Portland, Oregon 97225

Chaos in the Laboratory

and 13 Other Science Projects Using the Apple® II

David L. Vernier, Editor

Contributing Authors:

J. Roger Bacon
Peter Bruecken
Mark C. Helmlinger
Satoshi Kino
David L. Vernier

Our *How to Build a Better Mousetrap and 13 Other Science Projects for the Apple® II* has been used by thousands of students and teachers as an introduction to laboratory interfacing. *Chaos in the Laboratory* has fourteen new computer-related science projects. Each project includes background information and detailed step-by-step instructions for assembling an electronic circuit. Sample programs for use with the projects are included on two disks. Suggestions are given for extending the project by improving the programs, adding to the hardware, or doing further experiments.

Projects:

- Thermistor Temperature Probe
- Quiz Show
- Computer Vision
- Colorimeter
- A-to-D Converter
- Light Meter
- Barometer / Pressure Sensor
- Magnetic Field Sensor
- Smart Battery Charger
- D-to-A / Digitized Sound
- Stepper Motors
- Speed of Sound
- Ultrasonic Range Finder
- Chaos in the Laboratory

Published by:

Vernier Software
2920 S.W. 89th Street
Portland, Oregon 97225
(503) 297-5317

Price: $25.95

ISBN 0-918731-46-1